ESD-TR-68-87

ESD-TR-68-87
ESTI FILE COPY

Technical Note 1968-10

M. C. Crocker
E. H. Swenson

Synchronous Satellite Stationkeeping Simulation

14 May 1968

Prepared under Electronic Systems Division Contract AF 19(628)-5167 by

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Lexington, Massachusetts

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology, with the support of the U.S. Air Force under Contract AF 19(628)-5167.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LINCOLN LABORATORY

SYNCHRONOUS SATELLITE STATIONKEEPING SIMULATION

M. C. CROCKER
E. H. SWENSON

Group 63

TECHNICAL NOTE 1968-10

14 MAY 1968

LEXINGTON MASSACHUSETTS

ABSTRACT

A program has been written to simulate the east-west stationkeeping of a synchronous satellite. Different ways of implementing the thrust sequence of rocket motors and solar sails are discussed.

Accepted for the Air Force
Franklin C. Hudson
Chief, Lincoln Laboratory Office

TABLE OF CONTENTS

NOMENCLATURE

a	=	area of the sail
C	=	velocity of light
$\vec{F}$	=	component of solar force perpendicular to the sail vector
$F_{a\lambda}$	=	average annual force on the satellite by radiation pressure in the longitudinal direction
F_r	=	force on the satellite in the radial direction
F_θ	=	force on the satellite in the colatitude direction
F_λ	=	force on the satellite in the longitudinal direction
$J_{n,m}$	=	coefficients of the tesseral harmonics
$J_{n,o}$	=	coefficients of the zonal harmonics
m	=	mass of the satellite
P_m	=	Legendre polynomials
r	=	distance from the center of the earth to the satellite
R	=	radius of the earth
$\vec{R}$	=	radius vector
S	=	solar constant = 1.39×10^3 watts/m^2
$\vec{S}$	=	sail vector
$\vec{S}_N$	=	normal to the sail
$\vec{T}$	=	orbital tangent vector
$\vec{U}$	=	sun vector
V	=	earth's potential
α	=	solar absorptivity
θ	=	colatitude of the satellite
λ	=	longitude of the satellite
λ_{nm}	=	longitude of the n and mth harmonic
μ	=	mass of the earth x gravitational constant
ω	=	earth's rotational rate
I_{sp}	=	specific impulse

SYNCHRONOUS SATELLITE STATIONKEEPING SIMULATION

I. INTRODUCTION

The motion of a satellite with a period of one sidereal day is perturbed by the gravitational attraction of the sun and moon, radiation pressure and the tangential component of the earth's gravitational geopotential. A Synchronous Satellite Simulation program (SSS) was written to study these effects and to determine the feasibility of maintaining a satellite near a specified synchronous position by automatically firing rocket motors and by solar sailing. The different parts of the SSS program are described in the Appendices.

II. UNDERLYING THEORY TO THE COMPUTER SIMULATION

A. Forces Acting On the Satellite

The influence of the gravitational attraction of the sun and moon contribute to a small diurnal oscillation in the longitude of the satellite of the order of 0.5 degrees[1] at the most. A typical daily longitude variation has been calculated by Molitor and Kaplan[2] and is shown in Fig. 1. However, the sun and moon's gravitational attraction does cause the orbit normal to precess[3] so that the inclination changes by about 0.85 degrees per year.

Radiation pressure will change the eccentricity of the orbit a small amount for satellites with ordinary area to mass ratios that are symmetric about a plane that contains the center of mass of the satellite and is perpendicular to the equatorial plane. By making the satellite not symmetric, it is possible to East-West stationkeep a synchronous satellite by radiation pressure forces. Methods for doing this are presented in a paper by Black et al[4].

B. Orbital Mechanics

The motion of a satellite is determined primarily by the earth's gravitational potential function

$$V = -\frac{\mu m}{r}\left[1 - \sum_{n=2}^{\infty} J_{n,o}\left(\frac{R}{r}\right)^2 P_n^o(\cos\theta) + \sum_{n=2}^{\infty}\sum_{m=1}^{\infty} J_{n,m}\left(\frac{R}{r}\right)^n P_n^m(\cos\theta)\cos(\lambda - \lambda_{n,m})\right] \quad (1)$$

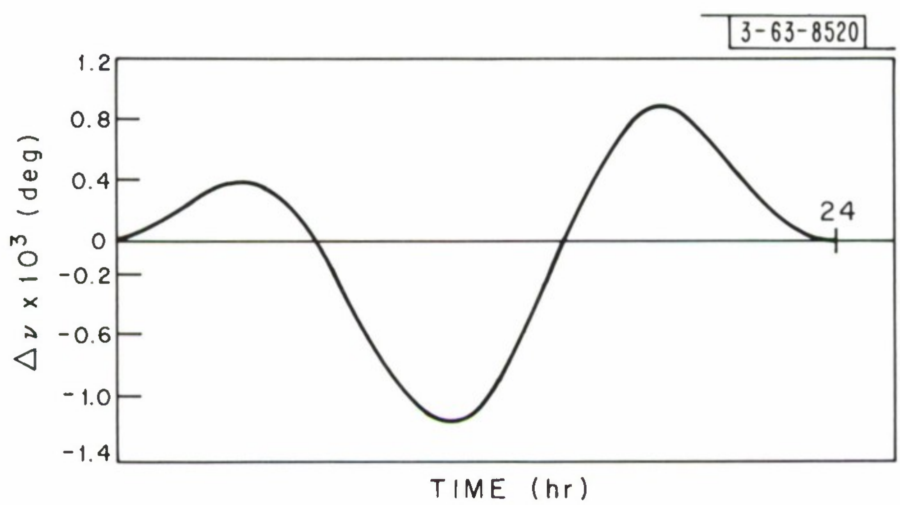


Fig. 1. Typical daily longitude variation of a 24-hour satellite due to solar and lunar perturbations.

For the purposes of studying east-west stationkeeping the effect of the gravitational attraction of the moon, sun and planets was neglected. The SSS program has the option of neglecting terms in the potential function greater than order two or three. The inaccuracies in measuring and producing surfaces with specified reflective properties produce uncertainties in the forces acting on the satellite which mask the above effects.

The motion of a 24-hour satellite is best studied by using an earth fixed system of coordinates rotating at the earth's angular rate ω.[5,7] The kinetic energy of the satellite expressed in this rotating system is

$$T = \frac{1}{2} m \left[\dot{r}^2 + r^2 \dot{\theta} + r^2 \sin^2 \theta (\dot{\lambda} + \omega)^2 \right] \tag{2}$$

Let us denote the forces on the satellite produced by rocket thrusters or radiation pressure devices as F_r, F_θ, F_λ. Now by using Lagrange's equations and retaining terms to the second order, the following equations of motion result.

$$\ddot{r} - r\dot{\theta}^2 - r \sin^2 \theta (\dot{\lambda} + \omega)^2 = -\mu/r^2 + 3\mu J_{20} R^2 (3 \cos^2 \theta - 1)/2r^4$$

$$+ 9\mu J_{22} R^2 \sin^2 \theta \cos 2(\lambda - \lambda_{22})/r^4 + F_r/m \tag{3}$$

$$\ddot{\theta} = \frac{3\mu J_{20} R^2}{2r^5} \sin 2\theta - \frac{3\mu J_{22} R^2}{r^5} \sin 2\theta \cos 2(\lambda - \lambda_{22})$$

$$+ \frac{(\dot{\lambda} + \omega)^2}{2} \sin 2\theta - 2\dot{r}\dot{\theta}/r + F_\theta/mr \tag{4}$$

$$\ddot{\lambda} = \frac{6\mu J_{22} R^2}{r^5} \sin 2(\lambda - \lambda_{22}) - 2 \operatorname{ctn} \theta \dot{\theta} (\dot{\lambda} + \omega) + F_\lambda/mr \sin \theta \tag{5}$$

$$- \frac{2\dot{r}(\dot{\lambda} + \omega)}{r}$$

When the force terms are set to zero F_θ, F_r, and F_λ the program has been found to simulate the motion of Syncom II to a high degree of accuracy as shown in Fig. 2.[6] The discontinuity in the curves is due to a rocket motor firing.

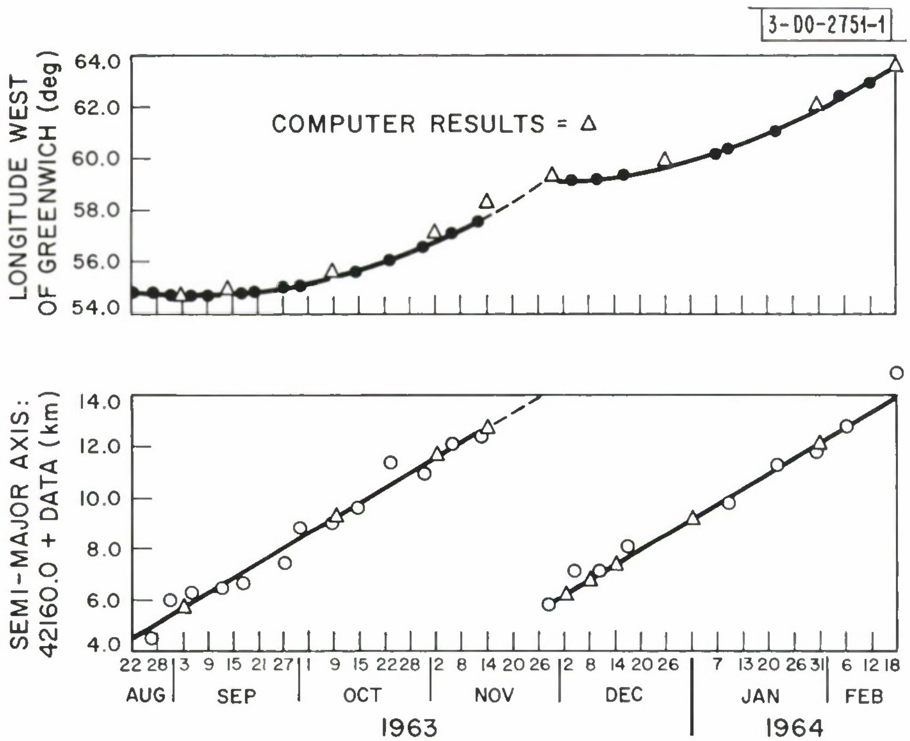


Fig. 2. SYNCOM II drift.

C. Longitudinal Perturbation Force

The longitudinal force per unit mass on a satellite due to the earth's geo-potential is given by taking the gradient of the potential

$$f_\lambda/m = -\nabla_\lambda V/m = \frac{1}{mr \sin\theta} \frac{\partial V}{\partial \lambda} = \frac{6J_{22}R^2\mu}{r^4} \sin\theta \sin 2(\lambda - \lambda_{22}) \tag{6}$$

For an equatorial orbit $\theta = 90^\circ$ and

$$f_\lambda/m = -5.568 \times 10^{-8} \sin 2(\lambda - \lambda_{22}) \text{ newtons/kilogram} \tag{7}$$

The stable equilibrium points are located at the positions $(\lambda - \lambda_{22}) = 90^\circ$ and 270°.

If a satellite were placed at these points, the satellite would remain there or oscillate about them with a small amplitude. No stationkeeping apparatus would be required if the satellite were placed at these points and the portion of the earth "visible" to the satellite was satisfactory. The stable equilibrium points are located off the western coast of South America and over the Indian Ocean. However, if the satellite is used for communications between the United States and Europe or the United States and the Far East the satellite will require stationkeeping apparatus. The visibilities of satellites placed for communication between the United States and Europe and between the United States and the Far East are shown in Fig. 3.

III. STATIONKEEPING BY SOLAR SAILING

The instantaneous radiation pressure forces, on the sail of a satellite, which has its plane in the north-south direction are given by the following vector equations, assuming there is no angular dependence of the solar absorptivity. These equations hold for any orbit.

$$F_\lambda = \frac{Sa}{c}\Big(h\,(2-\alpha)\,(\vec{S}_N \cdot \vec{U})^2 \,|-\vec{S}_N \cdot \vec{T}| - s\alpha\,|\vec{S}_N \cdot \vec{U}|\,(\vec{U}\cdot\vec{S})\sin[\cos^{-1}(-\vec{S}_N \cdot \vec{T})]\Big) \tag{8}$$

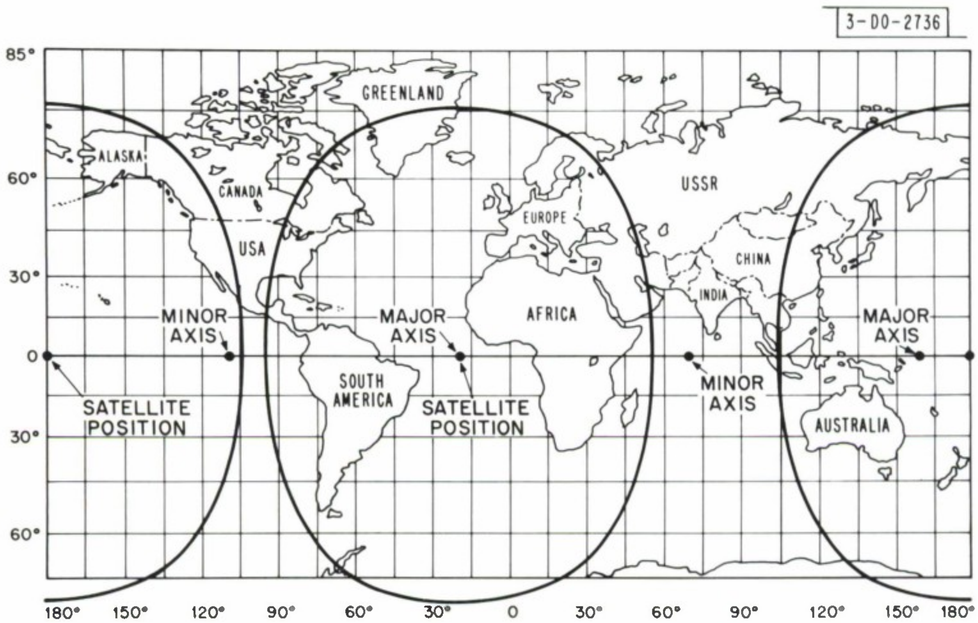


Fig. 3. Visibilities of satellites placed at stable equilibrium points.

$$F_r = \frac{Sa}{c}[g(2-\alpha)(\vec{S}_N \cdot \vec{U})^2 \sin[\cos^{-1}(-\vec{S}_N \cdot \vec{T})]$$

$$-f\alpha |\vec{S}_N - \vec{U}| |\vec{S}_N - \vec{T}| (\vec{U} - \vec{S})] \tag{9}$$

$$F_\theta = -\frac{Sa}{c}\alpha \cdot |\vec{S}_N \cdot \vec{U}| U_z \tag{10}$$

where h, s, g, f are +1 with the following exceptions:

$$h = -1 \text{ when } (\vec{F} \cdot \vec{T}) \leq 0$$
$$s = -1 \text{ when } (\vec{S} \cdot \vec{T}) \leq 0$$
$$g = -1 \text{ when } (\vec{F} \cdot \vec{R}) \leq 0$$
$$f = -1 \text{ when } (\vec{S} \cdot \vec{R}) \leq 0$$

A. Sail Type I

The simplest type of sail for a satellite that has one side oriented to the earth and one side oriented toward the north direction, is a sail whose plane is coincident with the radius to the satellite from the center of the earth and the north-south direction. (See Fig. 4) This sail has a low solar absorptivity on one side and a high solar absorptivity on the other. The thermal emissivity of both sides is approximately the same. Under these conditions the force resultant from thermal radiation from both sides of the sail cancels for a thin sail. The net force along the orbit is then due only to solar radiation pressure. The average annual force in the longitudinal direction is given by

$$F_{a\lambda} = \frac{\alpha_2 - \alpha_1}{4} \frac{Sa}{c} < \cos i > \tag{11}$$

where $<\cos i>$ is average annual effect of the sun's movement above and below the orbital plane of the satellite. If one assumes that the earth's orbit about the sun is circular and an equatorial orbit for the satellite $<\cos i> \approx 0.96$ and α_2 is the solar absorptivity of the sail facing in the direction of orbital

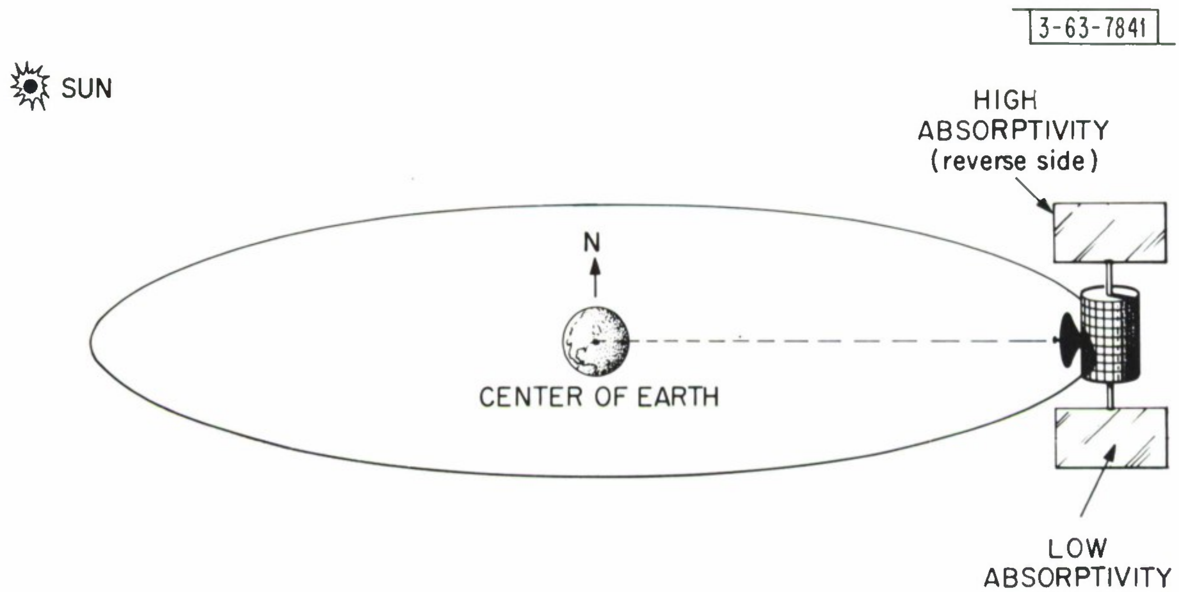


Fig. 4. Solar sail Type I.

motion and α_1 is solar absorptivity of the opposite face. This type of sail does not lend itself to being adapted to the stationkeeping methods to be discussed in Section IV.

B. Sail Type II

The second type of sail is similar to the first except both sides of the sail have the same highly reflective surface and can be rotated along the earth satellite line so that the plane of the sail goes from the north-south to east-west direction. The sail is left in the north-south portion for half the orbit starting (or ending) with the sun-earth line and the east-west position for the other half of the orbit. (See Fig. 5) The average annual force in the longitude direction is

$$F_{a\lambda} = \left[\pm \left(\frac{2 - \alpha}{4} \frac{Sa}{c}\right) < \cos i > - \alpha \frac{Sa}{2\pi c} < \sin 2i > \right] \qquad (12)$$

The positive sign is used when the north-south position is taken for half the orbit starting with the earth sun line and the negative sign is used when the north-south position ends with the earth sun line. This type of sail is more complicated than the first type because it requires a rotating mechanism and sensor logic to determine the location of the earth sun line. It does require less sail area than the first type. It is also adaptable to the previously mentioned methods of thrusting by rotating the sail into or out of position with respect to the earth-sun line.

C. Sail Type III

The third type of sail has both sides highly reflective. It rotates about the north-south direction continuously with a period of 12 hours. (See Fig. 6) The average annual force in longitudinal direction is

$$F_{a\lambda} = \frac{Sa}{\pi c} \quad \frac{2}{3} + 2\,(1 - \alpha)\,\sin 2\varphi < \cos i > \qquad (13)$$

where φ is the angle between the plane of the sail and the projection of the earth sun line on the equatorial plane when the satellite is crossing the projection of the earth sun line on the equatorial plane. This type of sail requires the least

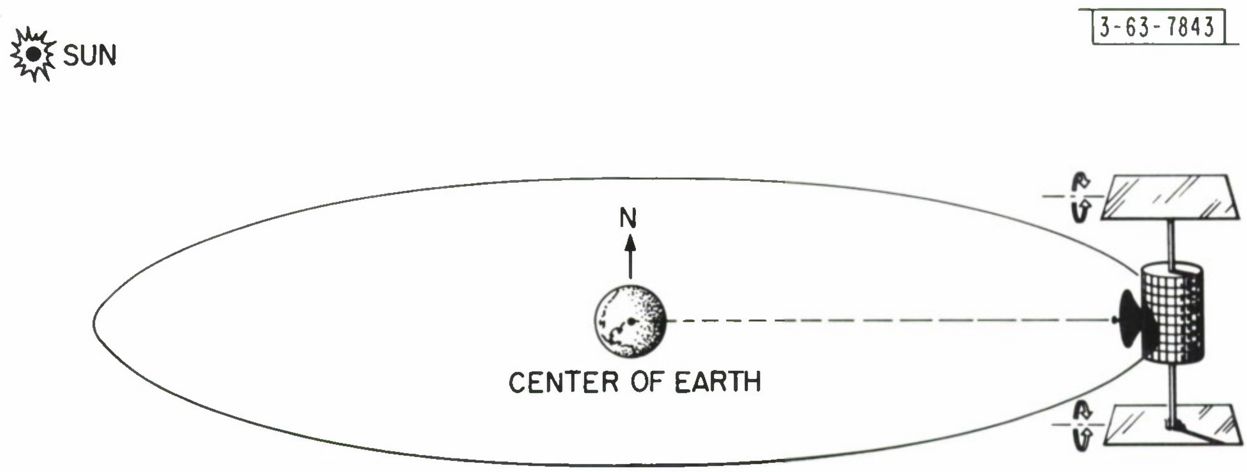


Fig. 5. Solar sail Type II.

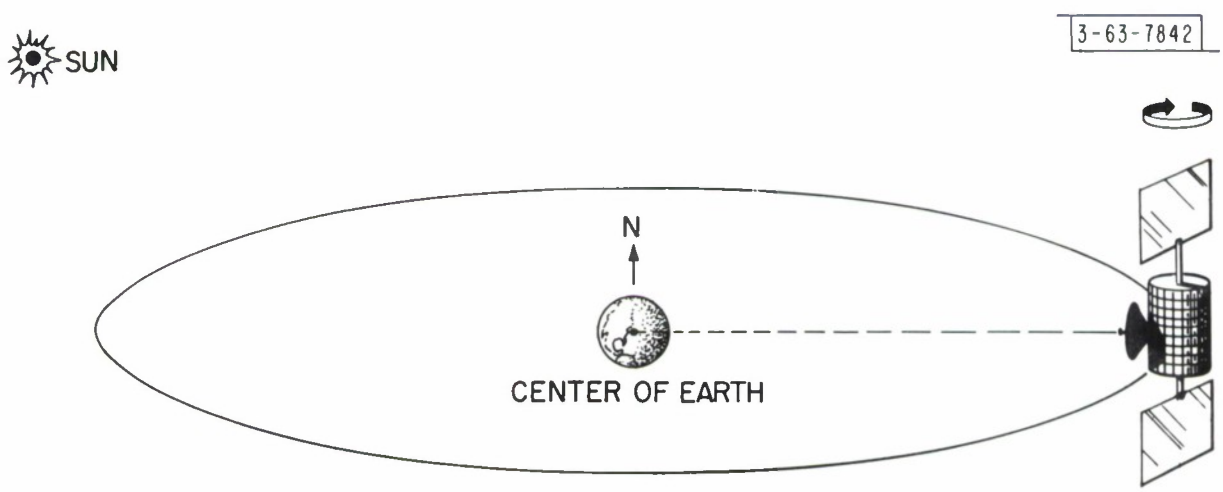


Fig. 6. Solar sail Type III.

amount of sail area for stationkeeping. If one sets $\varphi = 45^{o}$ and assumes a solar absorptivity of 0.9 for one side of the Type I sail and 0.05 for the other types of sails, the relative area needed to produce the same average annual force in the longitudinal direction for Type I, Type II and Type III sails is in the ratio of 3.84 : 1.70 : 1 respectively. A typical satellite with a mass of 100 Kg would require a Type III sail with an area of 1.47 square meters to stationkeep at the position of highest longitudinal force due to the earth's geopotential. By varying the value of φ it is possible to vary the longitudinal force along or against the orbital motion. Therefore, this type of sail is adaptable to any method of automatic stationkeeping.

D. Results of Computer Simulations

The computer simulation of stationkeeping a synchronous satellite by solar sailing revealed little difference between the three types of sail considered for stationkeeping. A typical stationkeeping simulation is shown in Fig. 7. In this simulation sail Type III was used with $\varphi = -45^{o}$ and $\alpha = .1$. The area of the sail was 1.35 square meters for a 100 Kg satellite. The satellite was stationkept near a position at which a maximum longitudinal force occurs. (296^{o} E longitude)

IV. AUTOMATIC STATIONKEEPING

A. Stationkeeping by Constant Force

Different methods of applying a tangential force to counteract the longitudinal perturbational force were simulated by the program. The simplest of these is to fire a rocket motor with the same value of longitudinal force that is acting upon the satellite. However, in practice the force cannot be exactly matched for each longitudinal position of the satellite. The method is limited if there is no control over the force level, once the satellite is placed in a longitudinal position within the longitude regions $45^{o} < \lambda - \lambda_{22} < 135^{o}$ and $225^{o} < \lambda - \lambda_{22} < 315^{o}$. If the force level is not exactly matched for a particular longitudinal position the satellite will undergo an oscillation about the longitude for which it is exactly matched. A simulation of an unmatched force level was done by the SSS program and is shown in Fig. 8. The satellite was initially at 291^{o} longitude and the force level was equal to the longitudinal force at 283.5^{o} longitude.

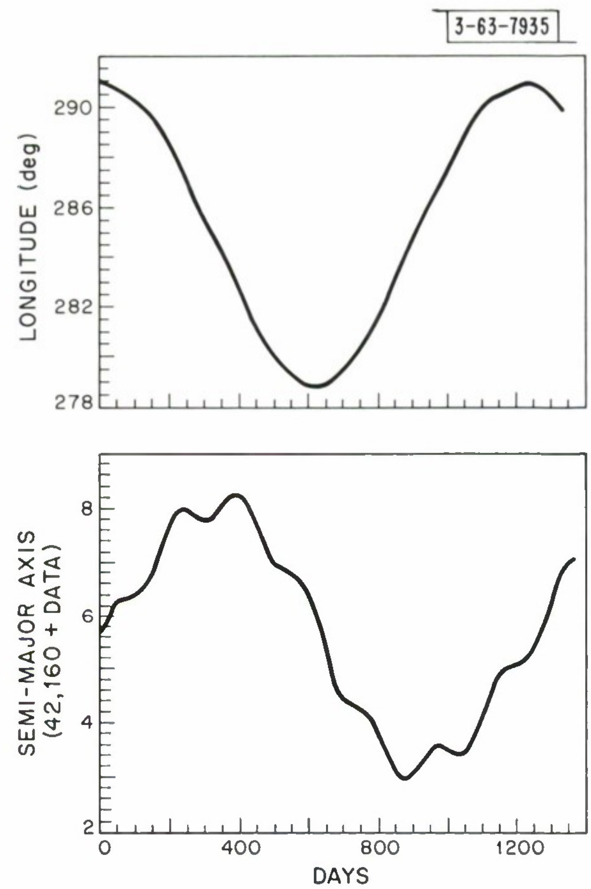


Fig. 7. Variations in semi-major axis and longitude versus time.

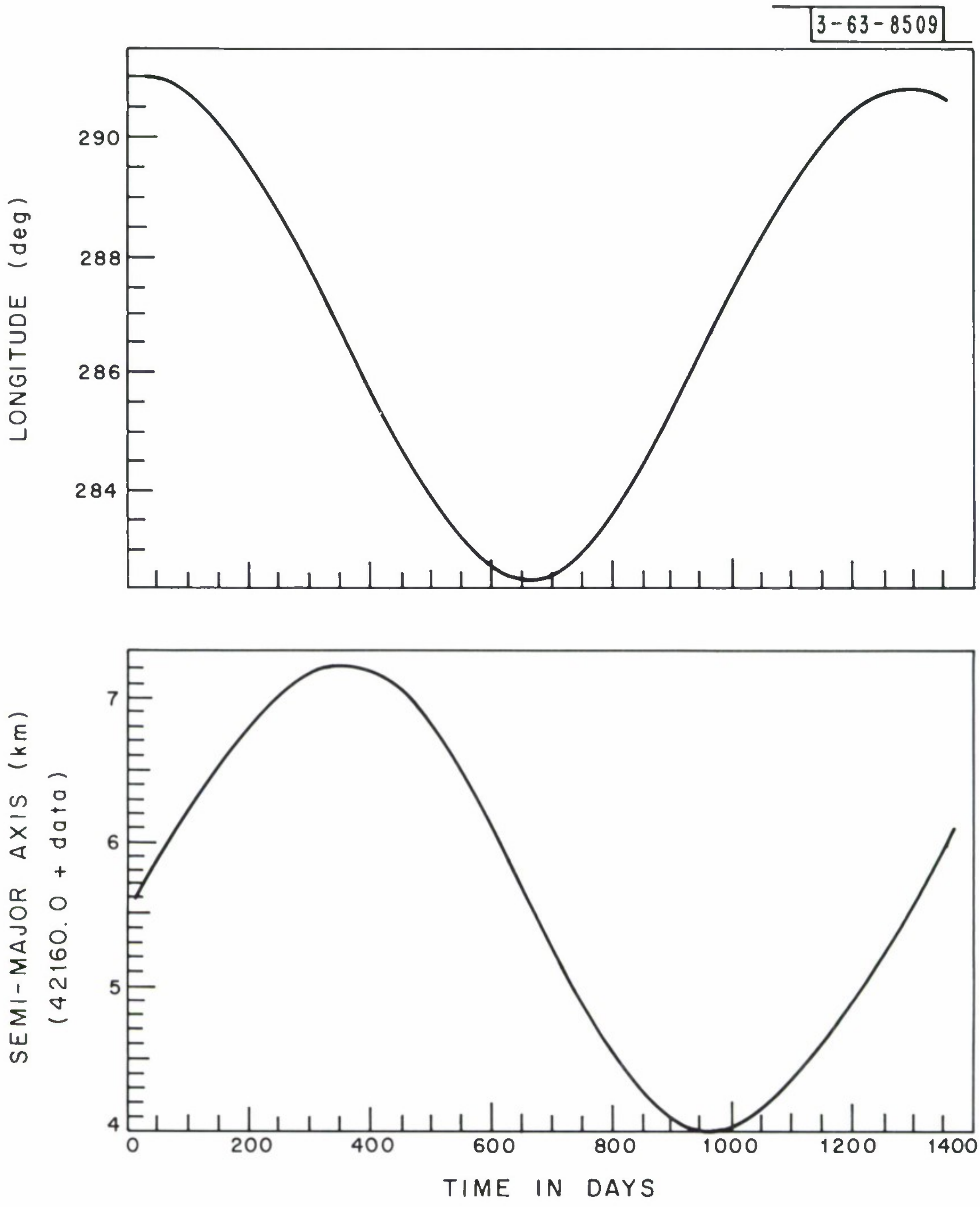


Fig. 8. Simulation of an unmatched force level.

B. Automatic Stationkeeping - Unidirectional Firing

Automatic stationkeeping was proposed by W. E. Morrow[8] in a memo dated 1 July 1965. A block diagram of the logic required by the satellite for internally computing its position and rocket firing times is shown in Fig. 9. This description does not describe the LES-6 stationkeeping logic.

It is the purpose of the coincidence logic to provide a pulse derived from fan beam sensors at a predetermined satellite position with respect to the earth and sun. The satellite may be assumed spinning so that the output from the sensors are in the form of pulses or for non-spinning satellite the output from the sensors may be chopped to produce the pulses. This coincidence pulse permits comparison between a running clock and a fixed clock in the decision logic. Almost any position in the orbit can be used for establishing the satellite's position where the optical sensors can view an illuminated portion of the earth. For definiteness let us pick the 6 a.m. position shown in Fig. 10.

The decision logic determines whether the satellite is early or late in arriving at the "6 a.m. position" each day. The device compares the coincidence signal with the output signal from the clock logic.

If the signal from the coincidence logic occurs in position A of Fig. 11, the satellite is early for the three days shown and no signal is sent to the thrust logic. If a signal from the coincidence logic falls in position B then a signal is sent to the thrust logic.

The thrust logic's purpose is to fire the rocket motor in the direction of the orbit tangent for a specified length of time near the 6 a.m. position. This is repeated daily until the satellite is "on station". The thrusting is then stopped until another signal is received.

The clock must have an accuracy of 5 in 10^{10}/day for 5 years and put out a signal (alarm) at 10 hr 12 min GMT each day. This signal is sent to the decision logic for further processing.

The purpose of the clock logic is to apply a daily correction to the clock to allow for the fact that the sun is ahead of or behind the clock according to

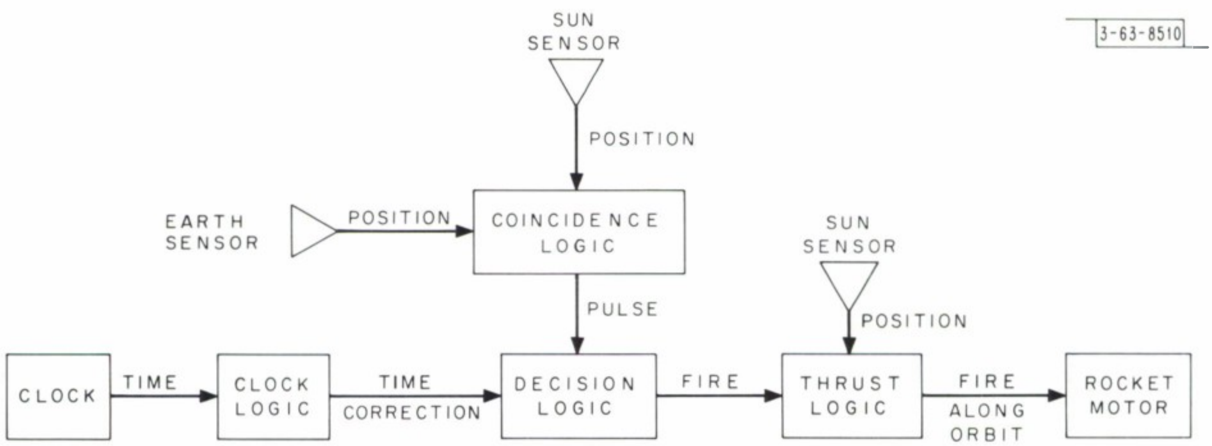


Fig. 9. Satellite logic for computing position and rocket firing times.

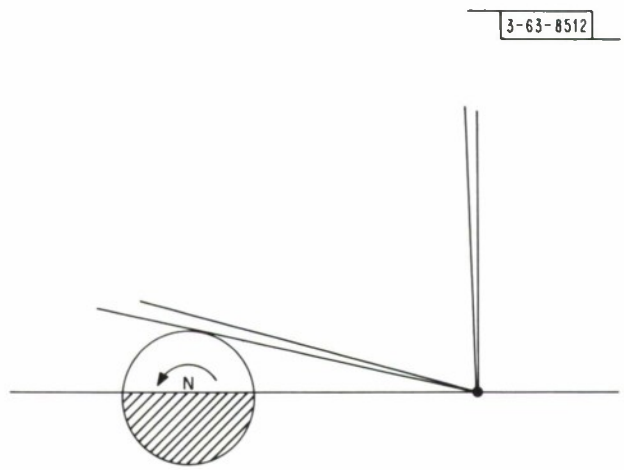


Fig. 10. 6 a.m. position of satellite in orbit.

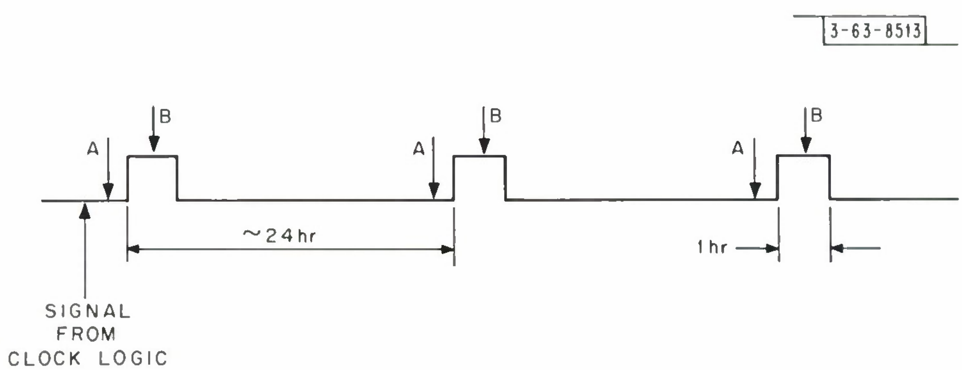


Fig. 11. Signal from clock logic.

the time of year. The correction that must be added to the time is shown in Fig. 12. The curve shown in Fig. 12 has to be approximated by a function that is easy to program into the satellite logic. The errors associated with the degree of exactness to which the clock logic will correct for the difference between apparent solar and solar mean time were accounted for from tables furnished by A. Braga-Illa for LES-6.

As an example of how the correction is applied, let us consider the correction at 15 January 1967. To do an exact calculation of the time to be added we would have to go through an iterative process. However, for our purposes the correction may be taken as the correction that occurs on January 15.0 which is 9.0 min. Therefore, the clock logic is to put out a signal to the decision logic which starts at 10 hr 21 min GMT and lasts for one hour. When the sun is ahead of the clock the correction has to be subtracted from 24 hours and the result added to the time of the previous days output.

The method is a little cumbersome in logic design. A better way to do it would be to shift 0 time base line so that all corrections become additions.

It is possible to do without the clock logic by using an optical sensor that senses the declination of the sun and relative position of the earth satellite line at the same time to correct for the actual longitudinal motion of the sun relative to the "mean" sun in the celestial sphere. The sensor can be visualized as having a cluster of pencil beams arranged in the familiar analemma found on some globes.[9] A sensor of this type is being constructed by C. Burrowes for use in LES-6.

C. Effects of Sensor and Clock Errors

There are errors associated with the measurement of longitude due to the beamwidth of the sensors and the degree of exactness to which the clock logic will correct for the difference between apparent solar and solar mean time. There are also errors induced because the LES-6 satellite spin axis is not aligned perfectly in the north-south direction. The source of error has been investigated by B. J. Moriarty.[10] The shape of the probability density function of the sum of the errors is not known. Therefore, a uniform probability

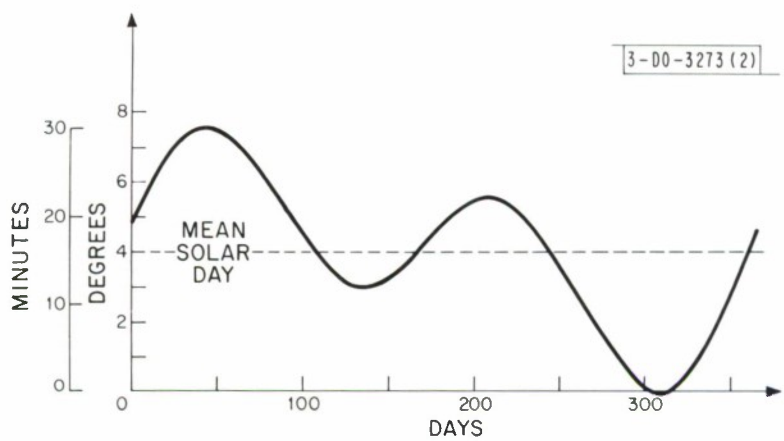


Fig. 12. Ephemeris correction vs. time of the year.

function with a width of one degree of longitude was assumed for use in the SSS program. The errors associated with the analemma sensor for LES-6 were separately accounted for from tables furnished by C. Burrowes.

The influence that the errors have on the motion of the satellite is shown in Figs. 13 and 14. Fig. 13 is the result of an SSS computer simulation of an automatic stationkeeping system using unidirectional firing with no errors present in determining longitude. Fig. 14 is the result of a simulation under the same conditions as the simulation of Fig. 13 except the clock logic errors and sensor errors are introduced. The errors do not significantly affect the range of longitudes over which the satellite may be stationkept. The errors in longitude sensing were less than 0.7 degrees due to clock errors and less than 0.5 degrees due to random sensor errors.

D. Automatic Stationkeeping Unidirectional Firing With Force Cut-Down

The best method of automatic stationkeeping so far proposed is that proposed by Roger Brockett.[11] This method uses the least amount of fuel and stationkeeps within the smallest range of longitude. As an example of how this method operates, let us assume the satellite is initially located at a longitude such that the earth's gravitational component along the orbit will cause the longitude of the satellite to decrease in time. The longitude of the satellite is allowed to decrease until it goes past a selected longitude, λ_o. At this point the rocket motors are fired for a specified length of time such that the impulse given per day is approximately five times the "impulse" given to the satellite by the earth's geopotential. This is the same as the method previously described. However, the longitude is sampled once per day and its value is stored in a memory bank in the satellite. If the longitude measured on a given day is greater than that measured the day before, the rocket impulse given per day is reduced until the satellite again passes the selected longitude, λ_o, at which time the thrusting is stopped. The amount that the impulse may be reduced is a variable that may be put in the input to the SSS program. A typical example of this stationkeeping method is shown in Fig. 15.

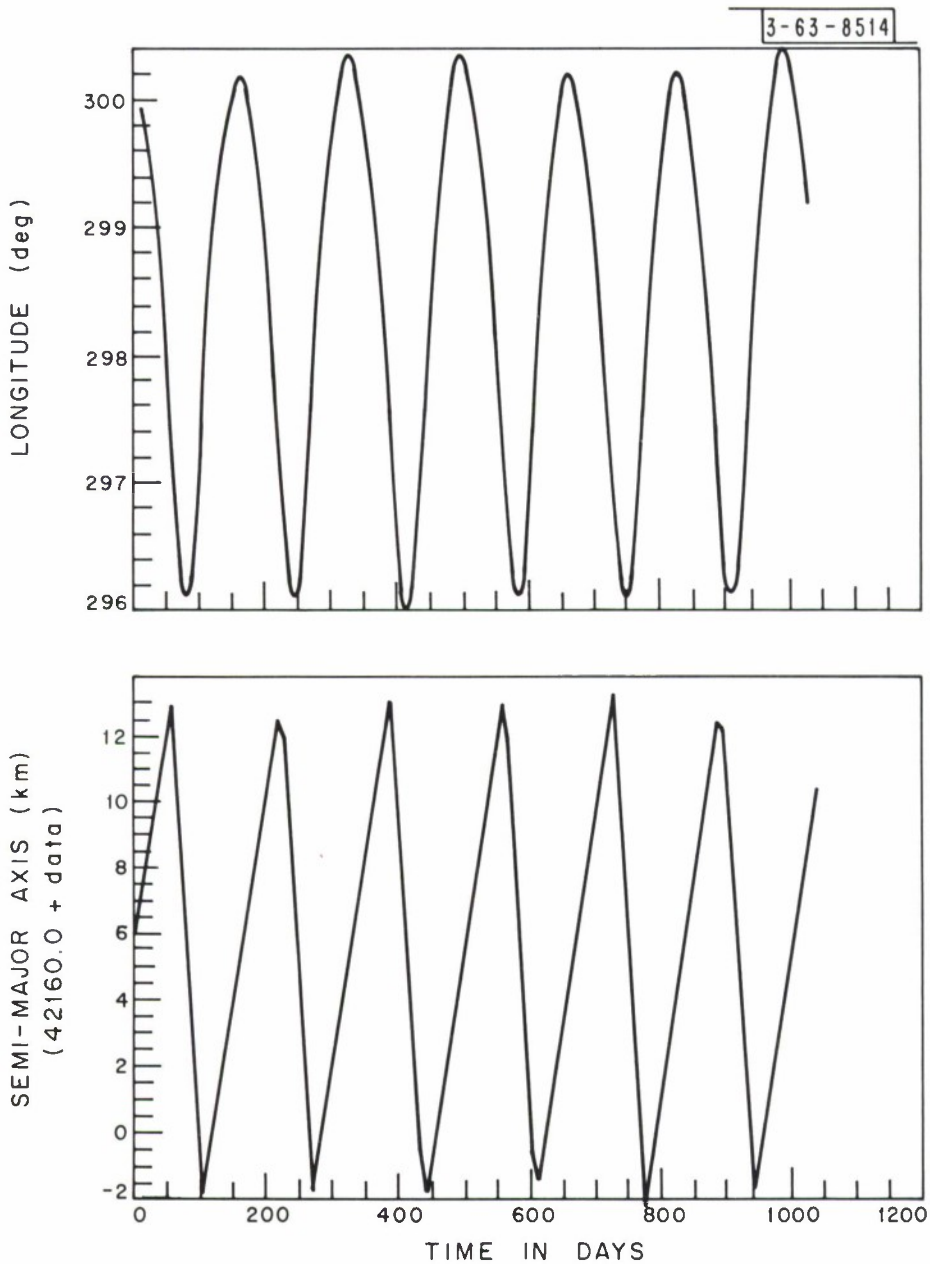


Fig. 13. Results of automatic stationkeeping with unidirectional firing with no errors in determining longitude.

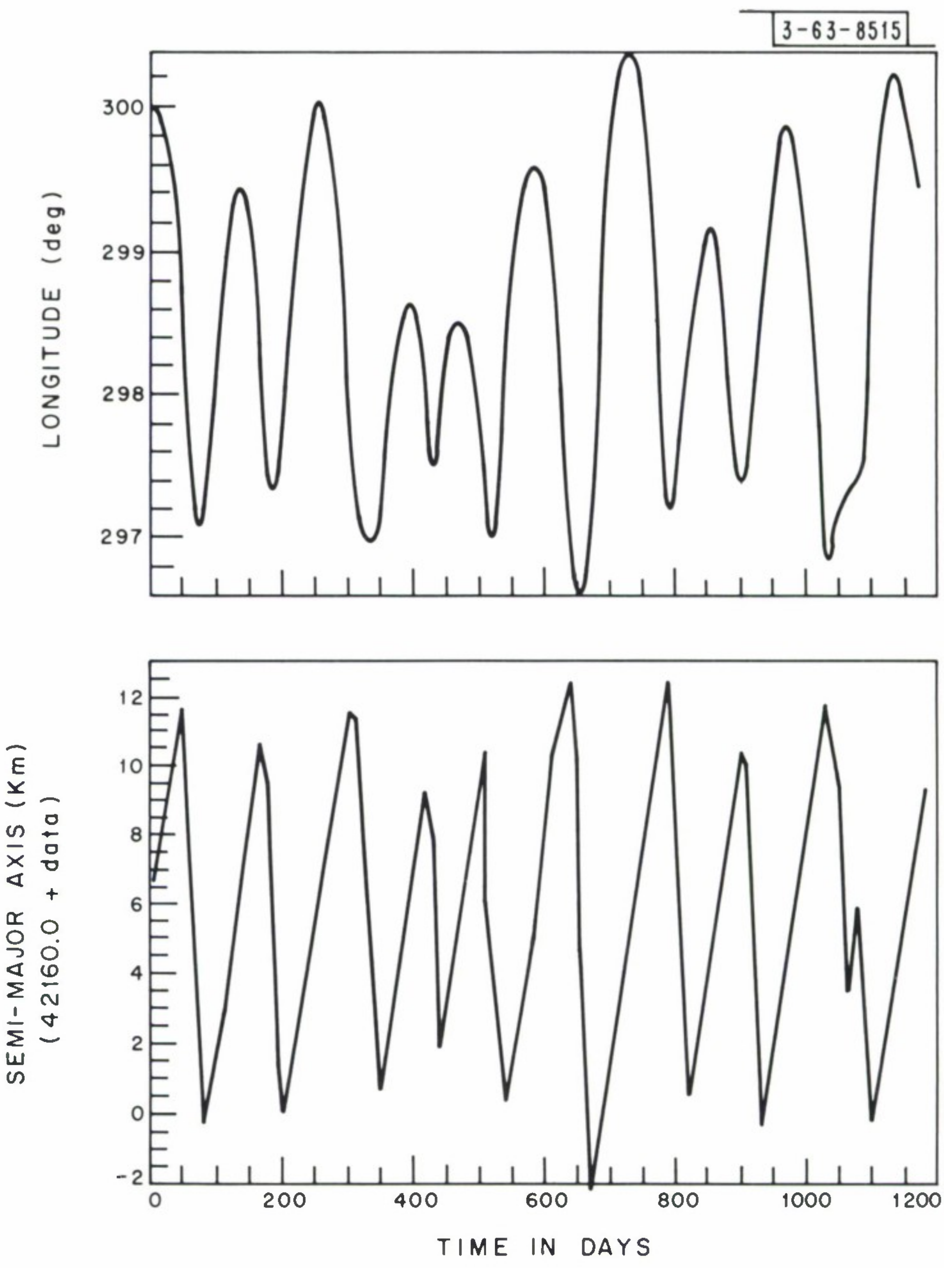


Fig. 14. Results of automatic stationkeeping with unidirectional firing with clock logic error and sensor errors introduced.

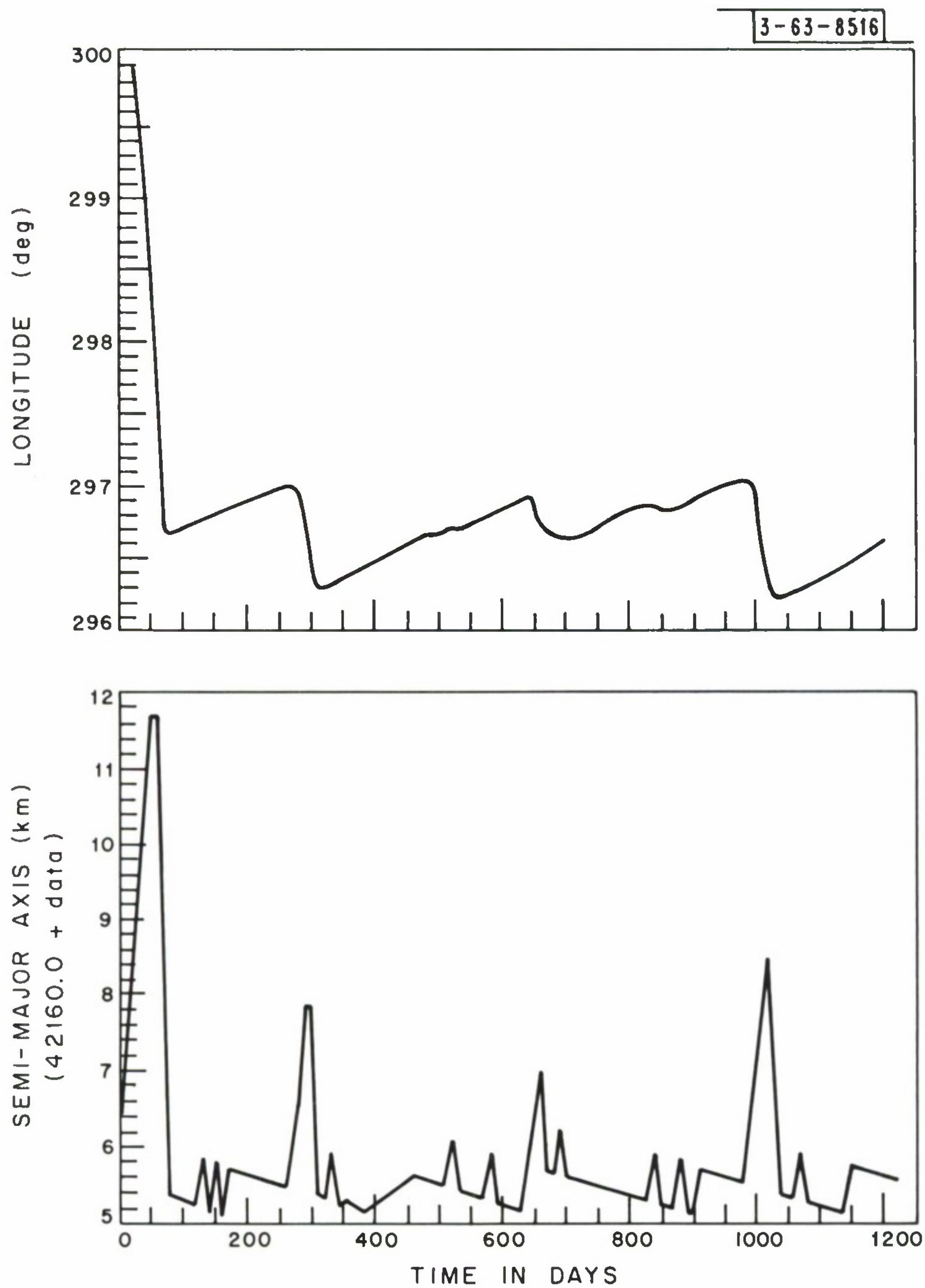


Fig. 15. Results of automatic stationkeeping with unidirectional firing with force cut-down.

E. Automatic Stationkeeping Bidirectional Thrusting

It is possible to stationkeep with a system which thrusts in one direction when the satellite exceeds a given longitude and in the other direction when the satellite is less than another given longitude. In other words, whenever the satellite has drifted out of a given band of longitude, thrusting is done to return the satellite to the band. This method of stationkeeping is not recommended for use with thrust systems that consume fuel at a low specific impulse because a lot more fuel is needed than with the methods previously mentioned. However, it is adaptable to solar sailing techniques where no fuel is used. A simulation of the bidirectional technique is shown in Fig. 16. This simulation was run under the same conditions as the unidirectional firing simulation of Fig. 13. The fuel consumption for stationkeeping a 132 Kg satellite for 1000 days was 3.85 Kg for a rocket system which has an $I_{sp} = 70$ for bidirectional thrusting. Under the same conditions only 0.86 Kg are needed for unidirectional thrusting.

Another method of bidirectional thrusting has been proposed by A. Braga-Illa. The logic of this method is explained in Appendix E describing subroutine Newfor of the SSS program. It is well suited for solar sailing techniques. A simulation of this method is shown in Fig. 17.

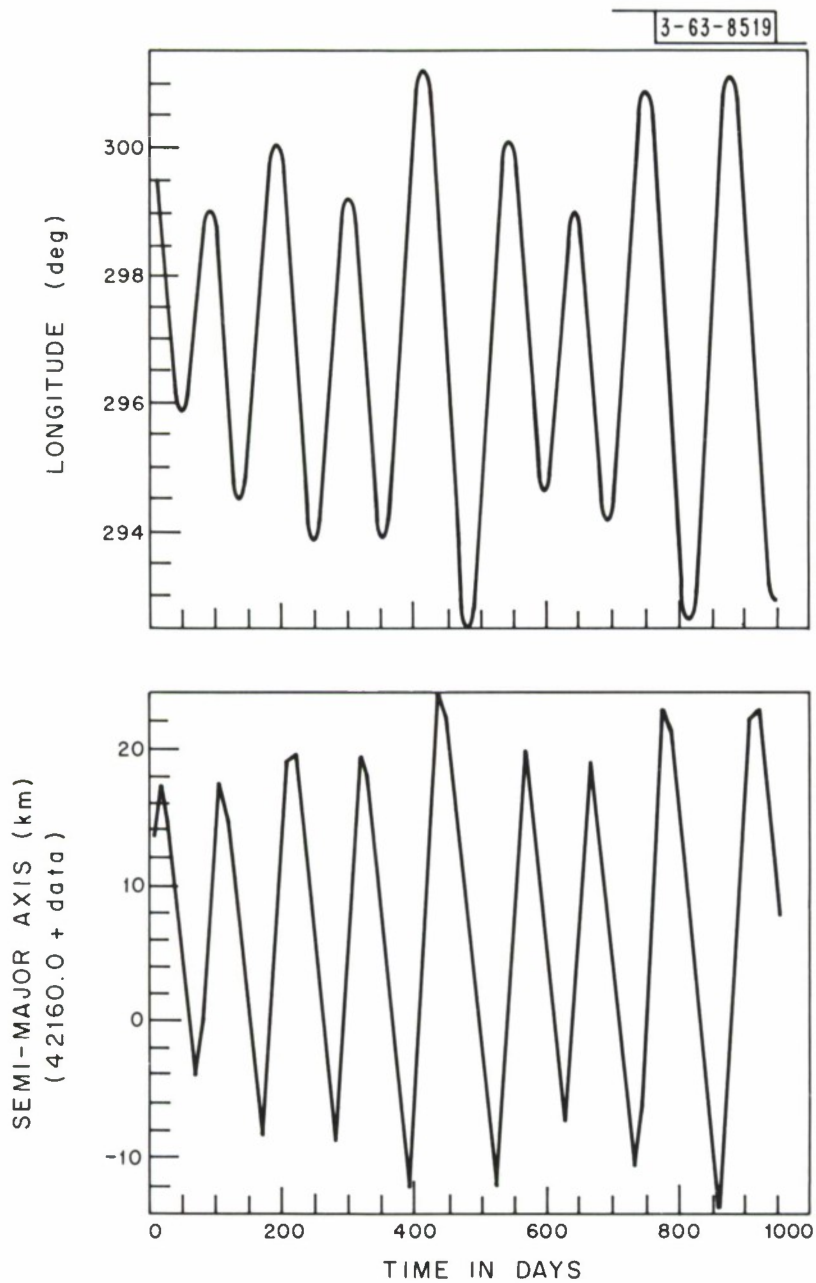


Fig. 16. Results of automatic stationkeeping using bidirectional firing.

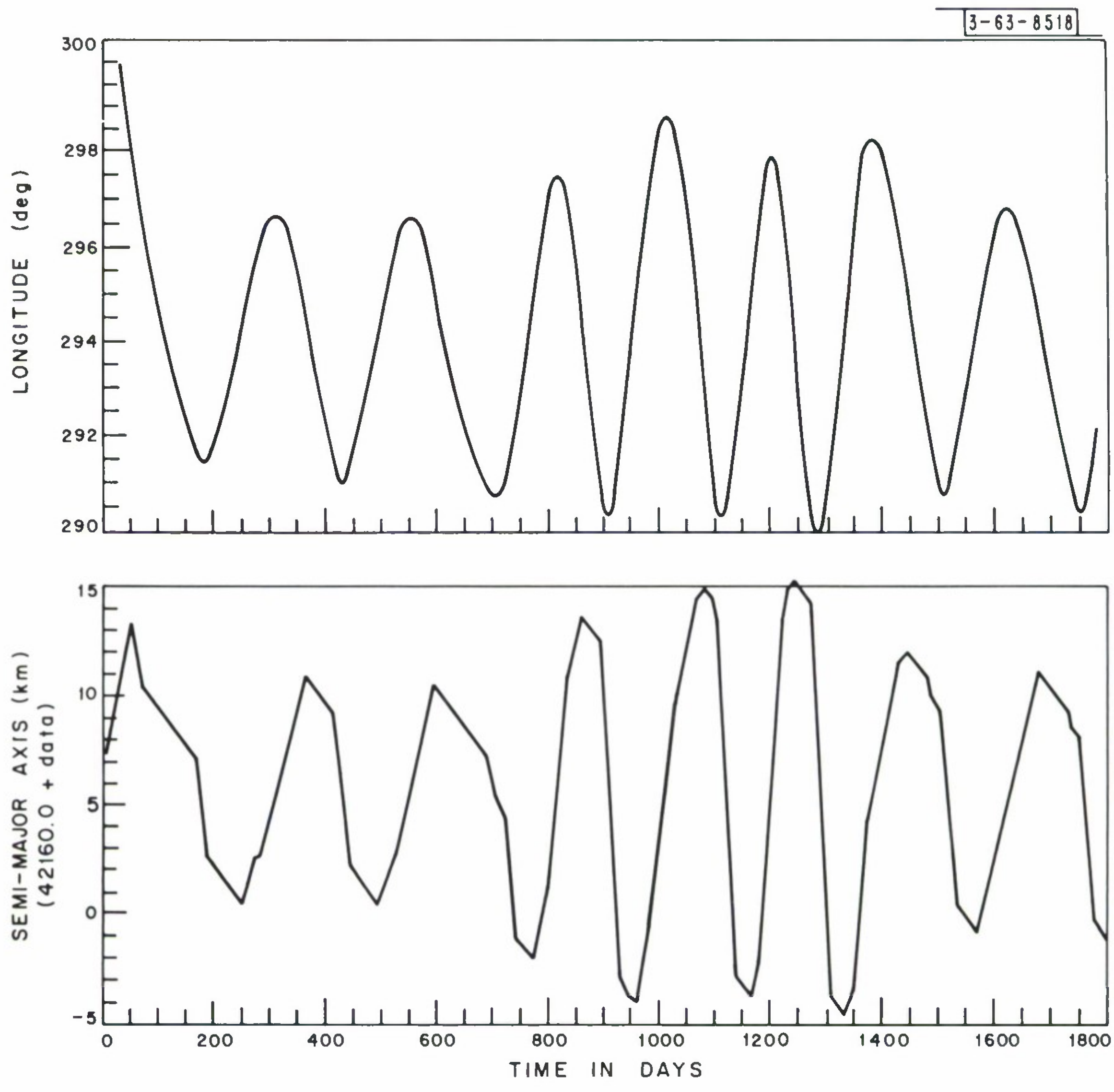


Fig. 17. Results of automatic stationkeeping using A. Braga-Illa's force scheme.

APPENDIX A

MAIN PROGRAM OF (SSS)

The main routine of the SSS program issues calls to the subroutines, solves the equations of motion, and produces the output of the program (Fig. A-1 and Appendix J). The initial section of comment cards gives a brief description of the program, its control variables and how to run it. The first executable statements are definitions of constants. The Main program calls GETINP to obtain the input data and control variables. After a sequence of housekeeping details -- moving variables to and from commons, changing variable types, and changing variables in degrees to radians -- the values necessary for the first entry into NEWFOR, FLAM, and CIND are initialized. These variables are initialized after the call to GETINP so that they are reinitialized at the beginning of subsequent sets of input data. The control variable I1 determines if the branch using second harmonics in the equations of motion or the branch using third harmonics is taken. In either case calls are issued to SSL and FLAM. SSL initializes CIND for solution of differential equations; FLAM returns the force in the λ direction. Certain logical paths determined by the input data return other variables to Main through COMMON (see Appendix D). After the call to FLAM, D2R, D2XLAM and D2THET are calculated. Both branches meet at statement 23, where three checks are made. First, is this time through the end of a day? Second, is this time through the end of ten days? Third, is this time through the end of amount of time asked for? If answer to all three questions are no, then it is end of one of the 86,400 times per day that the variables D2R, D2XLAM, and D2THET are calculated. At statement 6 the calls to FINDV, and DPNV provide T, R, THETA, XLAM, DR, DTHETA, and DXLAM for the following loop. Again a branch is taken based on I1. If it is the end of a day, the following are printed out.

TDAYS	days since beginning of this set of input data
R	radius of orbit
DR	change in radius in last loop of the day
TDEG	THETA in degrees

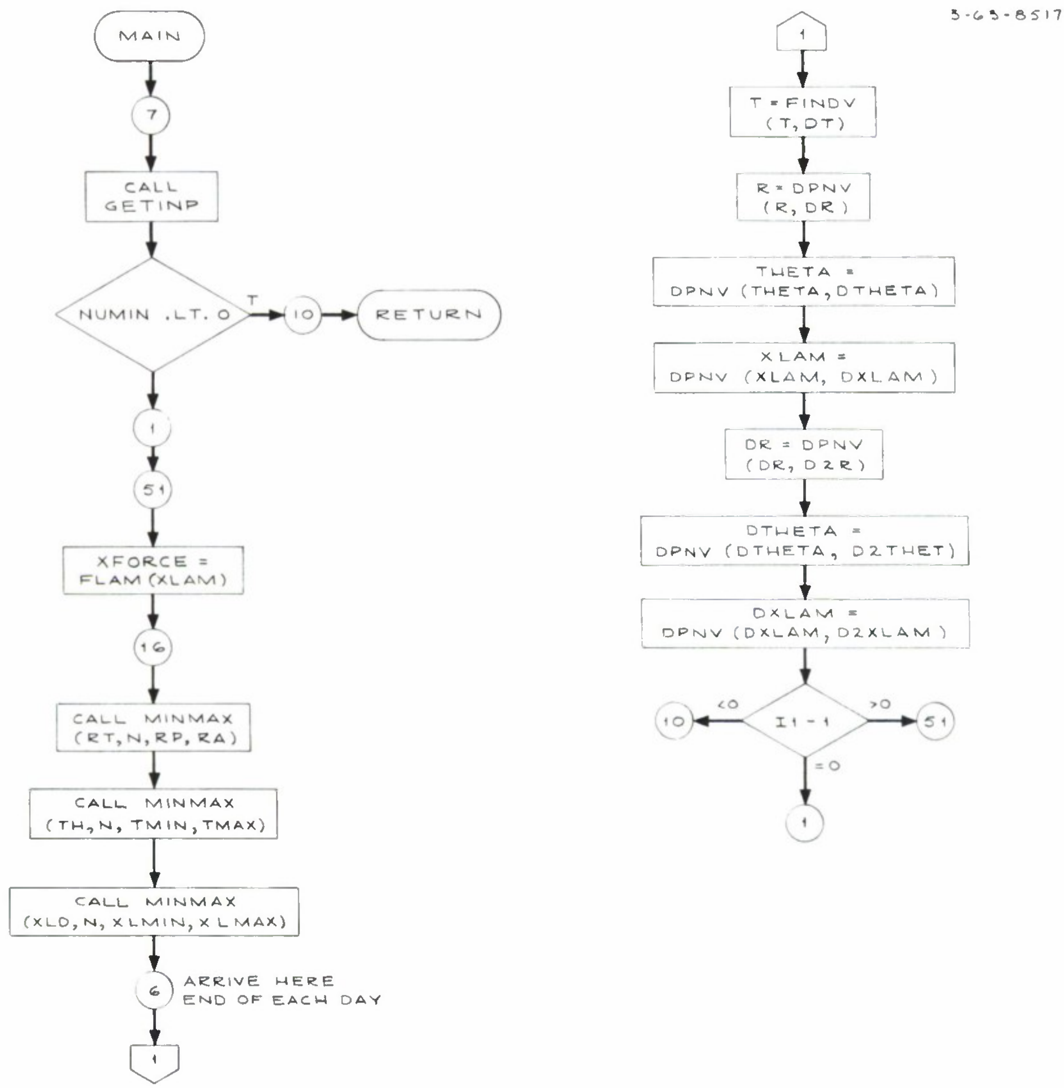
DTDEG	change in THETA in last loop of the day
XLDEG	XLAM in degrees
DXLDEG	change in XLAM in last loop of the day
XELONG	east longitude in degrees

After this printout a branch is made to question three. If the answer to question two is yes also, there precedes this printout a line of print containing the following averaged over previous 10 days.

A	radius
E	eccentricity of the orbit
XI	inclination
XIMID	minimum inclination
XLMAD	maximum inclination
DIFFER	XLMAD - XLMID

Every ten days there is also punched a card containing TDAY, A, and XLAM, to be used by a plotting program. After this there is a branch to question 3. If the answer to question 3 is yes, and the amount of output for one set of data is finished, the program branches to 7, CALL GETINP. The last set of data contains NUM = -1, which causes program to terminate normally. Each set of data, except the last, takes 1 1/2 - 2 1/2 hours of computer time.

MAIN PROGRAM FLOW CHART



APPENDIX B

The subroutine GETINP establishes the logical variables, all the input, and the modes of output. There are eight classes of integers that are used in conjunction with the logical variables: I1, I2, I3, I4, I5, I6, I7, I8. All the parameters are initialized in GETINP and then those that are to be changed are read in through a namelist setups. Using the logical parameters and the numerical parameters, the subroutine SETUP and its entry point RITOUT prepare the variable format for PRINT and/or PUNCH so that at the beginning of each output one may quickly review the particular set of parameters used. Control is then returned to the main program which then prints the titles for the output, prints and/or punches the orbital information every ten days.

I1 determines whether the calculations in the main program use second order harmonics or third order harmonics. I1 is set to 1 which selects the second order harmonics. If third is true, I2 then equals 2 and the third order harmonics are used. I2 controls the punched output. I2 is initially 1 which produces punched output. When no punched output is desired, I2 should read in as 2. The punched output is used by a separate program Z-PLOT which produces one frame of alphanumeric data and a second frame with 2 graphs longitude vs. days and semi-major axis vs. days. I3 controls the solar time correction and primitive uses of a solar sail. I3 is initialized as 1, no correction. If I3 equals 2 or linear is true, there is a linear solar time correction. If I3 equals 3 or LINRAN is true, there is a linear solar time correction with random error. If I3 equals 4 or perfect is true, the solar time is corrected perfectly. If I3 equals 5 or STEP is true, the solar time is corrected by a step-function. When I3 equals 6 or STPRAN is true, the step function correction is modified by calling the random error function. If I3 equals 7 or EPH is true, the solar time correction is done by an ephemeris sun sensor. When I3 equals 8 or EPHRAN is true, the ephemeris sun sensor correction is adjusted using the random error function. The remaining I3 values pertain to

the solar sail. When I3 equals 9 or SAIL1 is true, a fixed sail is used with a constant force along lambda. If I3 equals 10 or SAIL3 is true, a fixed sail scheme 3 is used. I4 determines which XJ22 and XLAM22 are to be used. The program is set up to use I4 equal to 2 which sets XJ22 at -1.700E-6 and XLAM at -19 degrees. There are no logical variables that go with I4. I5 establishes the force cutdown. Initially I5 equals 1, no force cutdown. When I5 equals 2, there is a force cutdown equal to the input number CUT. To obtain I5 equals 2, it can be read in directly or CUTDOWN can be read in as true. I6 and I8 are the more sophisticated uses of the solar sail. I6 equal to 1 or SAIL2 equal to true fixes the sail and allows one to also make a sun correction. I6 is initialized as 0. I6 equal to 2 or TACSAL is true the double precision function FLAM calls TACK and the tacked sail scheme is used. I8 flips the sail according to three schemes: I8 equal 1 or FLIPW equal to true fixes the sail through half the orbit and flips it off with the orbit; I8 equals 2 or FLIPAG true fixes the sail through half the orbit and flips if off against the orbit; I8 equals 3 or SAIL4 true flips the sail due to logic within the satellite. I7 determines if the orbit elements are averaged at the end of every ten days before they are punched. Initially I7 equals 2 and AVERAG is false, which loops the main program around the averaging sequence. When I7 equals 1 or AVERAG is true, the main program averages every ten days and punches the averaged figures.

The basic units for the parameters are kilometers, seconds, degrees. They are initialized as follows:

A	=	42165.5D0	radius of earth in kilometers
A1	=	0.9D0	solar absorptivity for solar sail
A2	=	0.1D0	solar abosrptivity for solar sail
ALONGR	=	0.0	force along the radius to the satellite
CONST	=	-4.45D-8	kilonewtons/kilogram
			= 1.6 * (factor = 5.6)/365.25/(86.4 * 6)
			sign is negative in second and fourth coordinates

CUT	=	0.0	
DATE	=	0.0	
E	=	0.0	
FORCCT	=	0.0	force along theta
FORLAM	=	1.0	force along lambda
IRUN	=	0	a counter so one can keep track of the various computer runs
ITHDAY	=	10	frequency in days of print and/or punch execution
NUM	=	6	number of impulses given (used in FLAM)
OFFSET	=	-90.0D0	angle by which sail can be offset in the TACK scheme
OL	=	0.0D0	argument of perigee for first orbit, in degrees
TEND	=	1800.0D0	number of solar days program is to run for
XDISP	=	70.0D0	specific impulse of thruster in seconds
XI	=	0.0D0	inclination of the orbit in degrees
XLAMD	=	300.0D0	degrees of longitude measure east of Greenwich
XLIMD	=	297.0D0	minimum angle for thrusting, simple scheme
XLIMXD	=	350.0D0	maximum angle for thrusting, simple scheme
XMASS	=	132.0D0	mass of satellite in kilograms

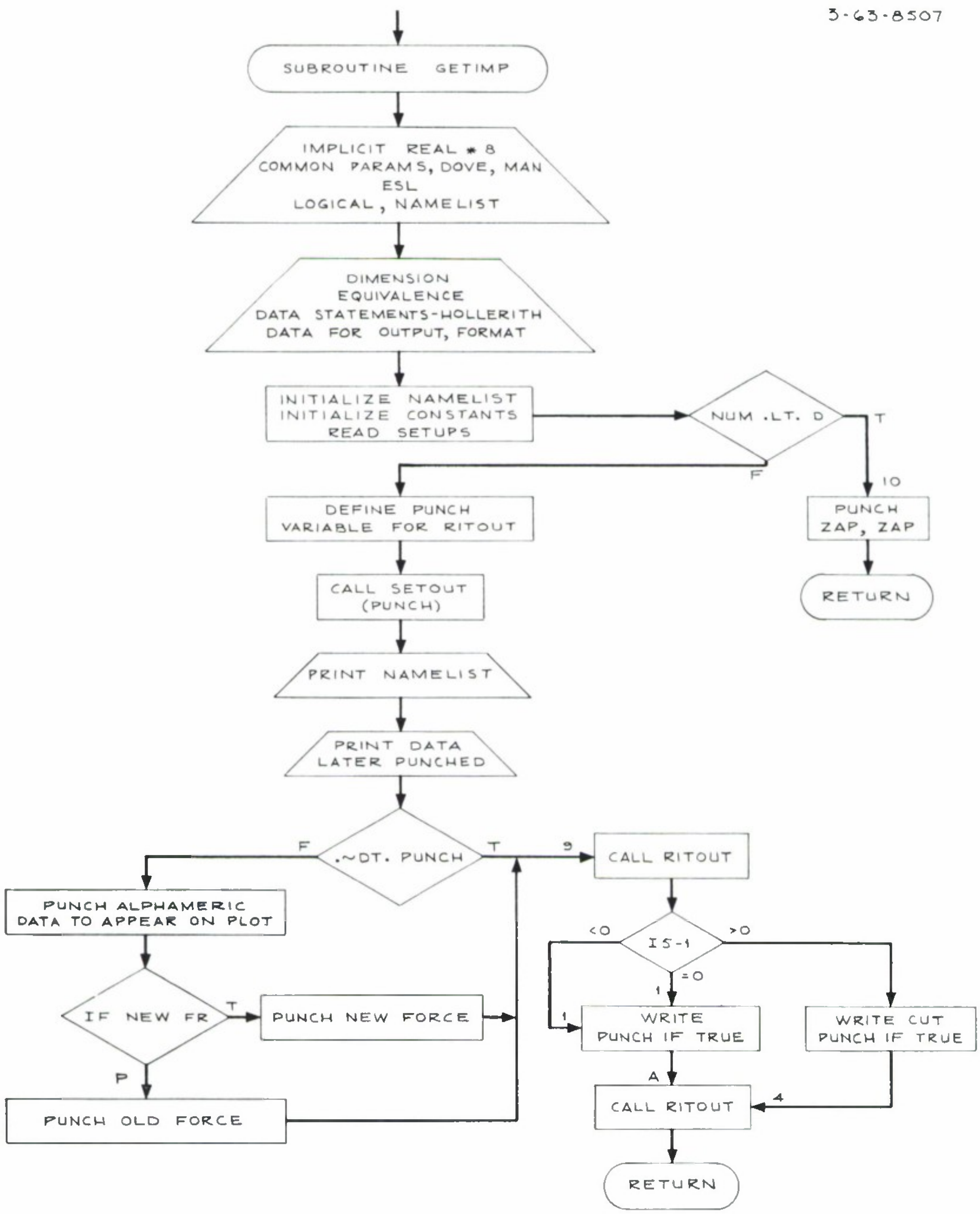
In addition there is a logical parameter, NEWFR, that has no numerical equivalent. NEWFR is initially FALSE. When NEWFR is true, the SUBROUTINE NEWFOR is called by FLAM and the magnitude of the impulse if modified accordingly.

To change any of the parameters or logical variables the NAMELIST/ SETUPS/ is used &SETUPS is on the first card beginning of column 2. The variables desired changes are then punched separated by commas. To set a logical variable to TRUE just use T for example: NEWFR = T would be

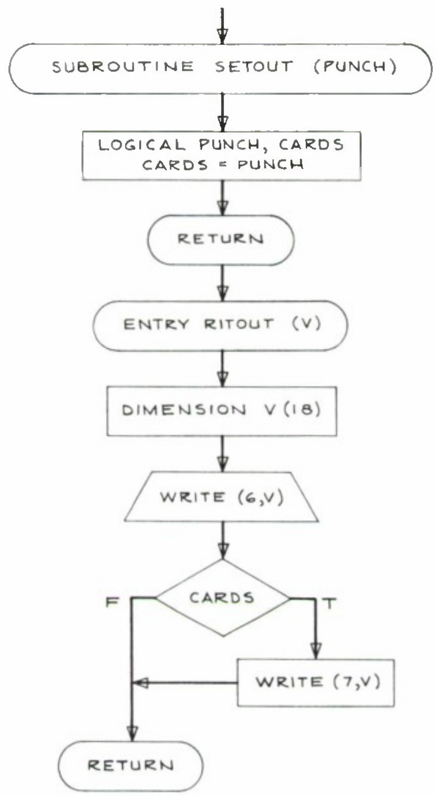
sufficient to set NEWFR true. Do not punch in column 73 or beyond. When using a second card begin in column 3. Do not follow the last variable with a comma; instead leave one or more spaces and punch &END. The set of cards with the SETUPS changes is then placed behind the //FT05F001 control card.

The parameters are printed and/or punched according to a variable FORMAT routine, RITOUT. RITOUT is an ENTRY point in SETOUT. The first call to SETOUT establishes whether or not to punch when RITOUT is called. RITOUT is called for each I variable. To add an I variable to the program, IY, where Y is an integer greater than 8, eight additions to GETINP are necessary. First a comment card to describe the IY variable and the X uses to which IY can be put. Secondly, it must be initialized. The remaining additions are to setup the call to RITOUT. There are X + 1 dimension statements as follows: Dimension VFIY(16, 6) and X statement like dimension XVFIYX(16). For each X there is also an EQUIVALENCE statement such as: EQUIVALENCE (XVFIXY(1), VFIX(1, Y). For each value IX can assume there must be a data statement in Hollorith form containing the alphanumeric data you desired printed and/or punched to describe the alternative used. You need to add IX to the 101 format statement, and one call to RITOUT, for example: Call RITOUT (VFIY(1, X)).

GETINP thus establishes the particular parameters and logical choices for one computer run, which takes approximately two hours on the IBM/360 with TEND = 1800.


3-63-8507
SUBROUTINE GETIMP
IMPLICIT REAL * 8
COMMON PARAMS, DOVE, MAN
ESL
LOGICAL, NAMELIST
DIMENSION
EQUIVALENCE
DATA STATEMENTS-HOLLERITH
DATA FOR OUTPUT, FORMAT
INITIALIZE NAMELIST
INITIALIZE CONSTANTS
READ SETUPS
NUM .LT. D
T
F
10
PUNCH
ZAP, ZAP
RETURN
DEFINE PUNCH
VARIABLE FOR RITOUT
CALL SETOUT
(PUNCH)
PRINT NAMELIST
PRINT DATA
LATER PUNCHED
.~DT. PUNCH
F
T
9
CALL RITOUT
PUNCH ALPHAMERIC
DATA TO APPEAR ON PLOT
IS-1
<0
>0
=0
1
1
IF NEW FR
T
PUNCH NEW FORCE
P
WRITE
PUNCH IF TRUE
WRITE CUT
PUNCH IF TRUE
A
4
PUNCH OLD FORCE
CALL RITOUT
RETURN

3-63-8508


SUBROUTINE SETOUT (PUNCH)
LOGICAL PUNCH, CARDS
CARDS = PUNCH
RETURN
ENTRY RITOUT (V)
DIMENSION V(18)
WRITE (6,V)
CARDS
F
T
WRITE (7,V)
RETURN

APPENDIX C

INTEGRATION FUNCTION AND SUBROUTINE

The double precision function CIND(K, A, B) and its related functions FINDV, DPNV, and SUBROUTINE SSL find the first and second derivatives and calculate new values for R, THETA, XLAM, DR, DTHETA, DXLAM, and T for each cycle through the main program. The call to SSL sets up the constants for SSL. FIND calls CIND with K = 0 which routes the control to the proper place to find a new T. For the other variables DPNV is used. In the computer GO TO statements only the first two solutions are used by this program. The sequence of four subprograms was written in another language outside Lincoln, so there are more choices than this program uses.

The integration formula used is the Adams four-point formula:

$$y_{n+1} = y_n + \frac{h}{24}(55y_n' - 59y_{n-1}' + 37y_{n-2}' - 9y_{n-3}') \qquad \text{(C-1)}$$

where $h = x_{n+1} - x_n$ is the constant incremental change of the independent variable x, y being a function of x.

Starting formulas are needed to obtain the first three points. The following set of formulas is used to obtain y_1; y_2, and y_3:

$$y_1^{(1)} = y_0 + hy_0' \qquad \text{(C-2)}$$

$$y_1^{(2)} = y_0 + \frac{h}{2}(y_1^{(1)\prime} + y_0') \qquad \text{(C-3)}$$

$$y_1^{(3)} = y_0 + \frac{h}{6}(2y_1^{(2)\prime} + y_1^{(1)\prime} + 3y_0') \qquad \text{(C-4)}$$

$$y_2^{(1)} = y_1^{(3)} + \frac{h}{2}(3y_1^{(3)\prime} - y_0') \qquad \text{(C-5)}$$

$$y_2^{(2)} = y_1^{(3)} + \frac{h}{12}\left(5y_2^{(1)\prime} + 8y_1^{(3)\prime} - y_0'\right) \quad \text{(C-6)}$$

$$y_3^{(1)} = y_2^{(2)} + \frac{h}{12}\left(23y_2^{(2)\prime} - 16y_1^{(3)\prime} + 5y_0'\right) \quad \text{(C-7)}$$

$$y_3^{(2)} = y_2^{(2)} + \frac{h}{24}\left(9y_3^{(1)\prime} + 19y_2^{(2)\prime} - 5y_1^{(3)\prime} + y_0'\right) \quad \text{(C-8)}$$

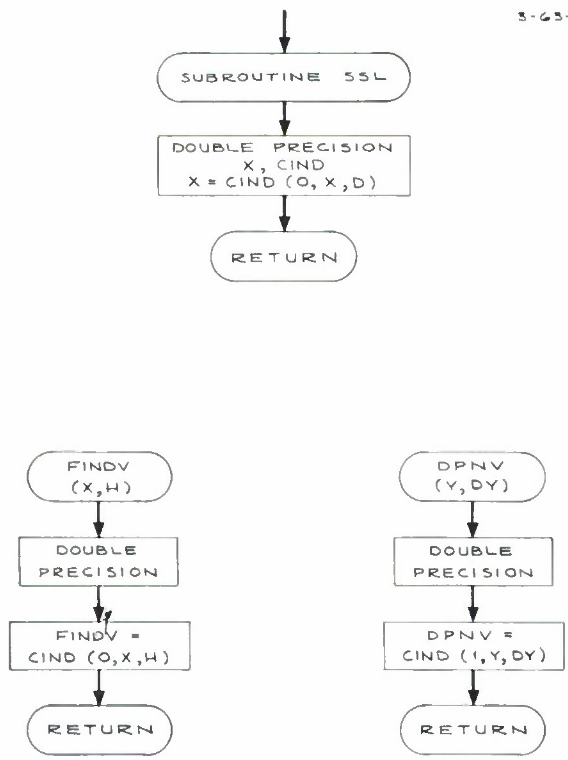
where the superscripts indicate the order of the approximation. The past values of the derivatives are stored in reserved blocks. For a more complete discussion of the starting equations, the reader is invited to contact the author. In particular, a substitution of equations (1) and (2) and similar expressions for their derivatives into equation (3) will show that equation (3) reduces to the first four terms of the Taylor series.

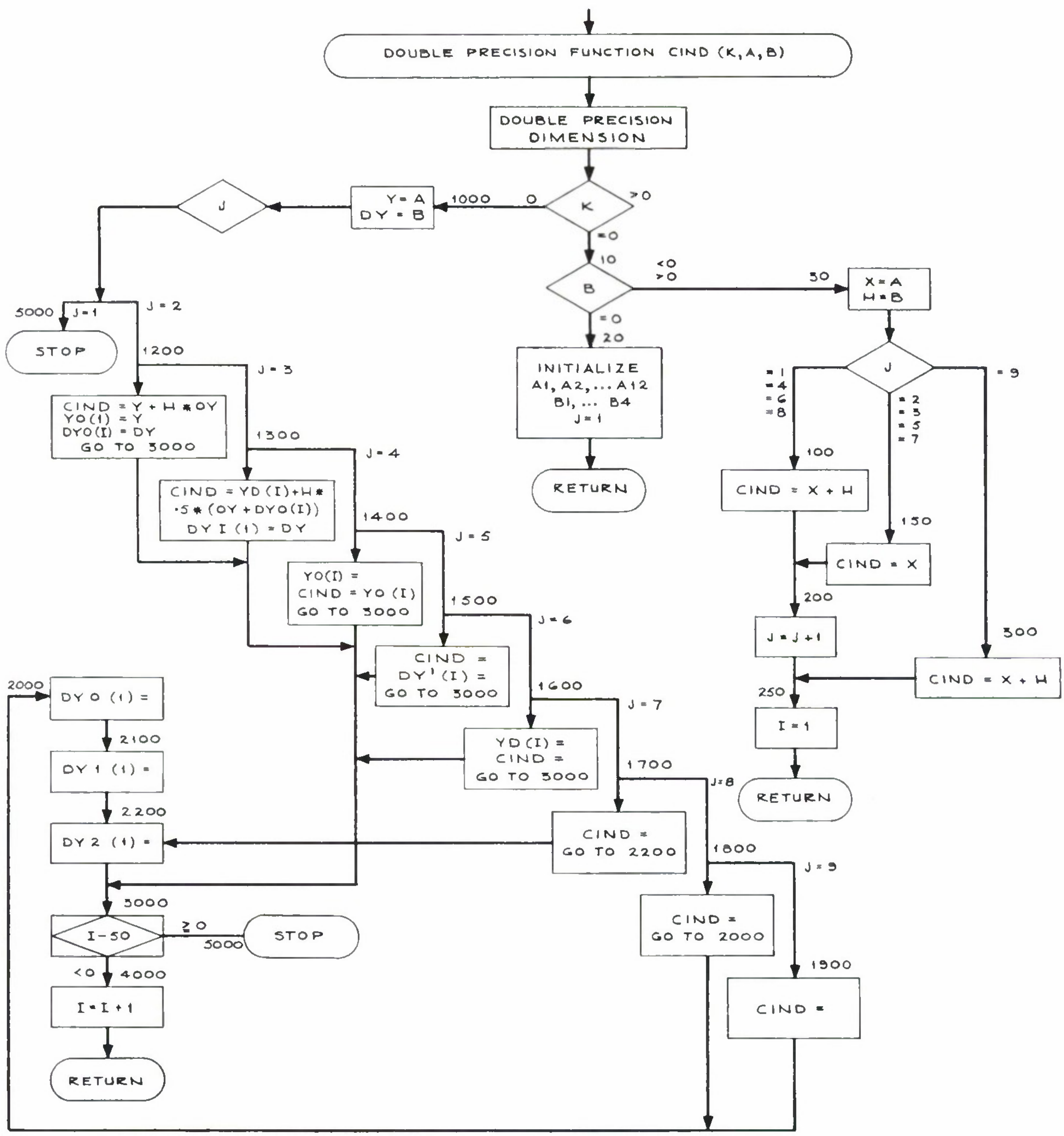
The truncation error of formula I is $\frac{251}{720} h^5 y^V(s)$, where $x_n < s < x_{n+1}$.

The accumulation of the increments $(\Delta y)_n = \frac{h}{24}(55y_n' - 59y'_{n-1} + 37y'_{n-2} - 9y'_{n-3})$ is performed in double precision to substantially reduce the round-off error.

Complete underflow-overflow testing is made. An underflow is corrected and a zero stored.

3-63-8505


SUBROUTINE SSL
DOUBLE PRECISION
X, CIND
X = CIND (0, X, D)
RETURN
FINDV
(X, H)
DOUBLE
PRECISION
FINDV =
CIND (0, X, H)
RETURN
DPNV
(Y, DY)
DOUBLE
PRECISION
DPNV =
CIND (1, Y, DY)
RETURN


DOUBLE PRECISION FUNCTION CIND (K,A,B)
DOUBLE PRECISION DIMENSION
K
>0
0
=0
1000
Y= A
DY = B
J
10
B
<0
>0
=0
30
X = A
H = B
20
INITIALIZE
A1, A2,... A12
B1, ... B4
J = 1
RETURN
5000
J = 1
J = 2
STOP
1200
J = 3
CIND = Y + H * DY
YO(1) = Y
DYO(I) = DY
GO TO 3000
1300
J = 4
CIND = YD (I)+H*
.5 * (DY + DYO(I))
DY I (1) = DY
1400
J = 5
YO(I) =
CIND = YO (I)
GO TO 3000
1500
J = 6
CIND =
DY' (I) =
GO TO 3000
1600
J = 7
YD (I) =
CIND =
GO TO 3000
1700
J=8
CIND =
GO TO 2200
1800
J = 9
CIND =
GO TO 2000
1900
CIND =
2000
DY O (1) =
2100
DY 1 (1) =
2200
DY 2 (1) =
3000
I-50
≥ 0
5000
STOP
<0
4000
I = I + 1
RETURN
J
= 1
= 4
= 6
= 8
= 2
= 3
= 5
= 7
= 9
100
CIND = X + H
150
CIND = X
200
J = J + 1
300
CIND = X + H
250
I = 1
RETURN

APPENDIX D

DOUBLE PRECISION FUNCTION FLAM (XLAM)

The double precision function FLAM(XLAM) computes a force, FLAM, simulating the force of the thruster or solar sail depending on the logic with which the program is run. FLAM is returned to the main program. The force is applied NUM times an orbit, controlled by the gate K. When K is positive the gate is closed returning both force and FLAM as 0. When K is negative control passes to label 8. The various routes for the logic depend on the force scheme used for the particular run in question. The variable 15 controls the cutdown; when 15 is 1 there is no cutdown. When 15 is 2, the force is cut by the factor CUT, as long as LLL is also positive. LLL is the control variable whose sign is determined by the satellites position within a specified band, delineated by XLIM and XLIMX. LLL is negative when XLAM is inside the band and positive when XLAM is outside the band. 13, for values 1-8, controls the type of clock error returned by CORR(TEL). From this an XTEMP is calculated. XLAM is the actual position of the satellite; XTEMP is where the satellite's logic thinks it is. The old force calculations use LLL, whereas the new force calculations use the subroutine NEWFOR. As long as the counter M is less than NUM, control returns to the main program with FLAM = FORCE. Once an orbit the fuel consumption, the total number of impulses, JFK, CORR the corrected longitude in degrees are printed out. The fuel consumption is kept as a running sum. After the fuel report is printed, on the last loop during which there is thrusting, K is set positive and the other counters are initialized for the next orbit.

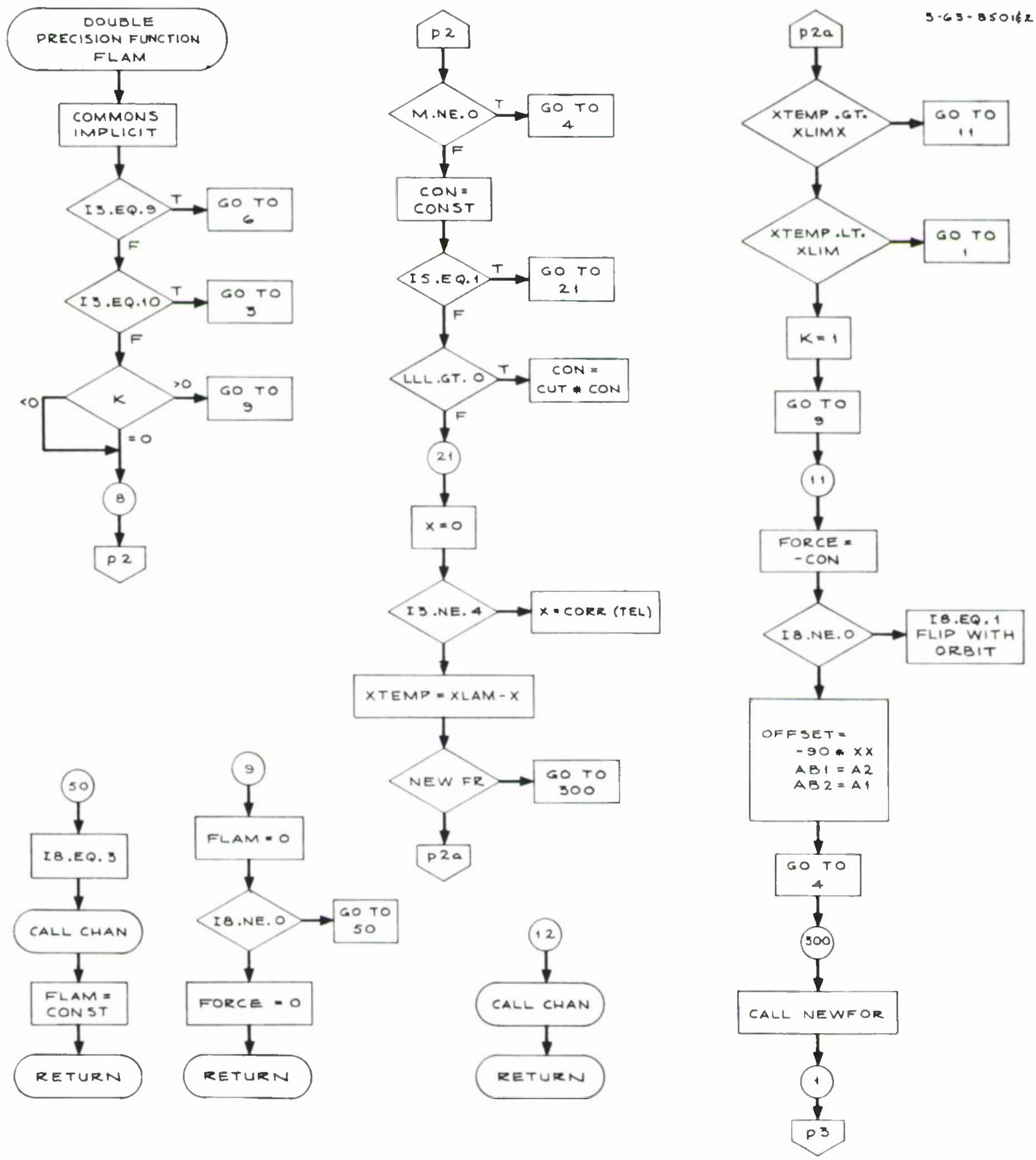
For the various schemes of solar sailing, there are deviations from the above possibilities. When K is positive and the sail is flipped off, FORCE = CONST and CHAN is called. When a simple sail is used FLAM = CONST and return. For this sail there is a constant force along λ. When a fixed sail is used, 13 = 10, FLAM = CONST and TACK or CHAN is called depending on 16.

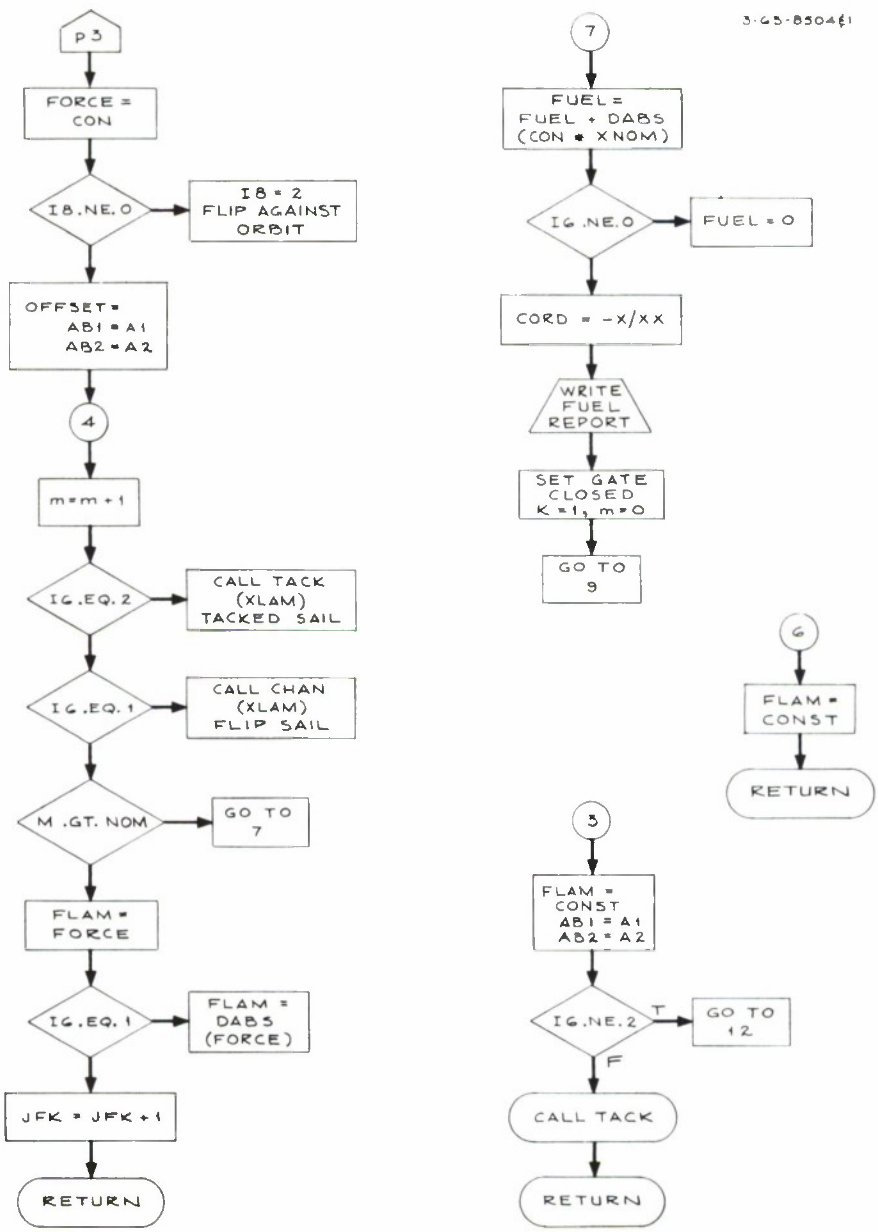
Tack calculates the forces in other directions as in addition to the force along λ the FLAM returns. Control returns to the Main program.

When K is negative (or zero) the force FLAM is calculated including the CUTDOWN or not. The first time through CON = CONST, XLAM is correlated by CORR(TEL). It is then determined if the modified XLAM lies within the band, if so FLAM = 0 and control is returned to the main program. If the true longitude is outside the desired stationkeeping range control passes to statement 1 or 11 within FLAM. If the satellite is too far east, FORCE = - CON, the dial can be flipped with the orbit if that scheme is used, or the sail can be offset -90° if that scheme is used. If the satellite is too far west, FORCE = CON, the sail can be flipped against the orbit, OFFSET can be 0°. In either case control passes to the same point 4, where K will return control for the rest of the orbit as long as the impulses are being given.

At statement 4 M dutifully counts the impulses. CHAN or TACK or neither is called according to 16. If M exceeds NUM, the fuel consumption report is issued. On all the sail schemes the fuel is set to zero. And the counters are reset for the next orbit as in the non-sailing schemes.

The double precision function FLAM(XLAM) returns to the main program NUM times an orbit the force along λ necessary to stationkeep within a specified range according to various force schemes using thrusters and solar sails.


5-63-8501¢2
DOUBLE PRECISION FUNCTION FLAM
COMMONS IMPLICIT
I3.EQ.9
GO TO 6
I3.EQ.10
GO TO 3
K
GO TO 9
8
p2
p2
M.NE.0
GO TO 4
CON = CONST
I5.EQ.1
GO TO 21
LLL.GT.0
CON = CUT * CON
21
X = 0
I3.NE.4
X = CORR (TEL)
XTEMP = XLAM - X
NEW FR
GO TO 300
p2a
p2a
XTEMP .GT. XLIMX
GO TO 11
XTEMP .LT. XLIM
GO TO 1
K = 1
GO TO 9
11
FORCE = -CON
I8.NE.0
I8.EQ.1 FLIP WITH ORBIT
OFFSET = -90 * XX
AB1 = A2
AB2 = A1
GO TO 4
300
CALL NEWFOR
1
p3
50
I8.EQ.3
CALL CHAN
FLAM = CONST
RETURN
9
FLAM = 0
I8.NE.0
GO TO 50
FORCE = 0
RETURN
12
CALL CHAN
RETURN


3-63-8504(1)
P3
FORCE = CON
I8.NE.0
I8 = 2 FLIP AGAINST ORBIT
OFFSET= AB1 = A1 AB2 = A2
4
m=m+1
I6.EQ.2
CALL TACK (XLAM) TACKED SAIL
I6.EQ.1
CALL CHAN (XLAM) FLIP SAIL
M .GT. NOM
GO TO 7
FLAM = FORCE
I6.EQ.1
FLAM = DABS (FORCE)
JFK = JFK + 1
RETURN
7
FUEL = FUEL + DABS (CON * XNOM)
I6.NE.0
FUEL = 0
CORD = -X/XX
WRITE FUEL REPORT
SET GATE CLOSED K = 1, m = 0
GO TO 9
6
FLAM = CONST
RETURN
5
FLAM = CONST AB1 = A1 AB2 = A2
I6.NE.2
T
GO TO 12
F
CALL TACK
RETURN

APPENDIX E

SUBROUTINE NEWFOR

Subroutine NEWFOR is an alternative method of calculating the force necessary to keep a satellite in a synchronous orbit using the bilateral thrusting method proposed by A. Braga-Illa.

A. Theory

Figure 18 shows a sample of longitude vs. time, and the regions ± A, ± B, ± C. The value of the force, con, at any point will be determined by which region we are in at that point and where we were before.

I In region C and -C, con = C_3 and $-C_3$ resp.
In region B and -B, con = C_2 and $-C_2$ resp.

For regions A and -A there is a complicated scheme for determining the value:

II If we get into region +A via +B or -A via -B, then con = O.
If +A via -A, con = C_1 or $-C_1$ resp.

III As soon as we get the sequence (A-A+A) or (-A+A-A) we start damping the force basically as follows. Assume we get the sequence +A-A+A and then the curve never leaves the +A, -A region.

Region:	A	B	+A	-A	$+\bar{A}$	-A	+A	-A	+A	-A	+A	-A
Force:	C_1	C_2	0	$-C_1$	$\frac{C_1}{2}$	$\frac{-C_1}{2}$	$\frac{-C_1}{4}$	$\frac{-C_1}{4}$	$\frac{C_1}{8}$	$\frac{-C_1}{8}$	$\frac{C_1}{16}$	$\frac{-C_1}{16}$
Region:	+A	-A	+A	-A	+A	-A						
Force:	$\frac{C_1}{32}$	$\frac{-C_1}{32}$	0	$\frac{-C_1}{32}$	0	$\frac{-C_1}{32}$						

where $\bar{A}$ is the region where we start damping. After (*) we do not damp but keep $\pm C_1/32$, 0 as long as we stay in +A-A and do not stray into B.

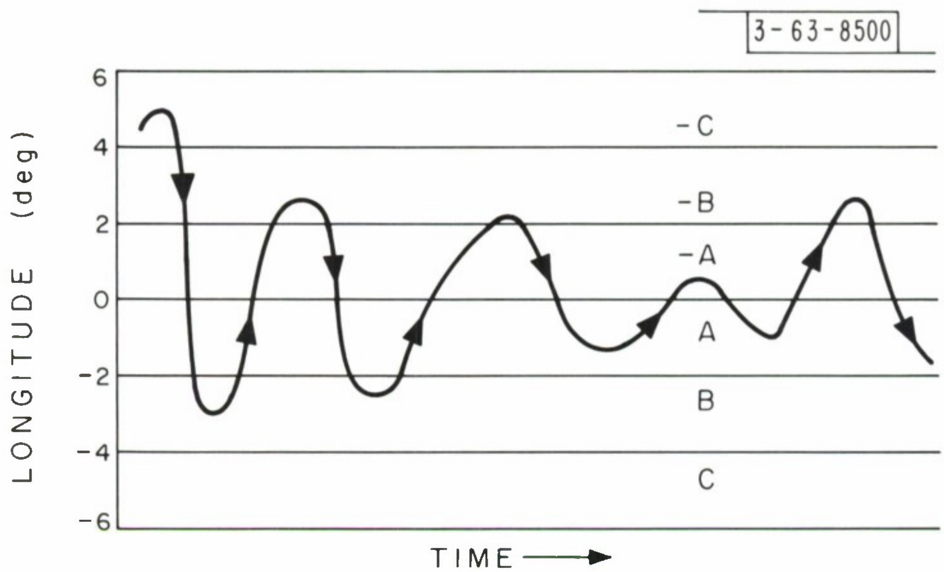


Fig. 18. Longitude vs time.

The most practical way to show the deviations from the basic scheme is via figure 19. Note the following:

(1) When we get into region C (see arrow 1) we start over in the sense that no damping occurs until we get the $\pm A \mp A \pm A$ sequence again (see arrow 2).

(2) When damping has started and we go into region $\pm B$, and then back into $\pm A$ we <u>keep</u> the damped value that we had before going into region $\pm B$, with the appropriate sign (see arrows 3, 4).

(3) When we reach the maximum damping stage $(\pm C_1/32, 0)$ and then go into region $\pm B$, upon re-entry into $\pm A$ the force = 0 by part II and if we then go into $\mp A$ from there, force = $C_1/32$ with the appropriate sign (see arrows 5, 6).

B. Data Cards

To specify that subroutine NEWFOR is to be used to calculate the force we must specify four quantities on the data cards:

NEWFR = T, C_1 = , C_2 = , C_3 =

where C_2, C_3 are the values assigned to the force in regions B and C respectively and C_1 is the non-zero undamped value of the force in region A.

3-63-8499

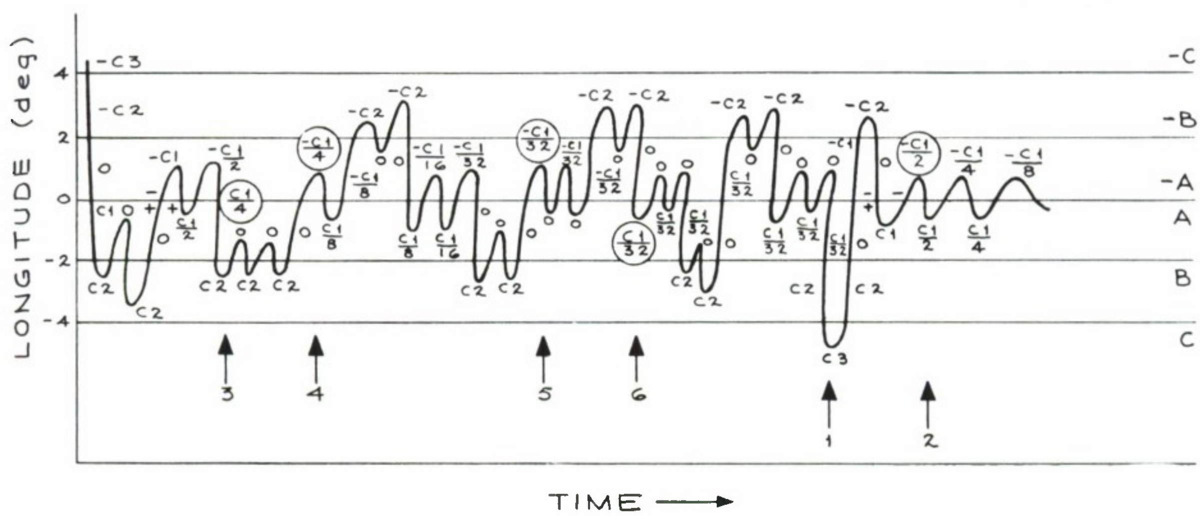
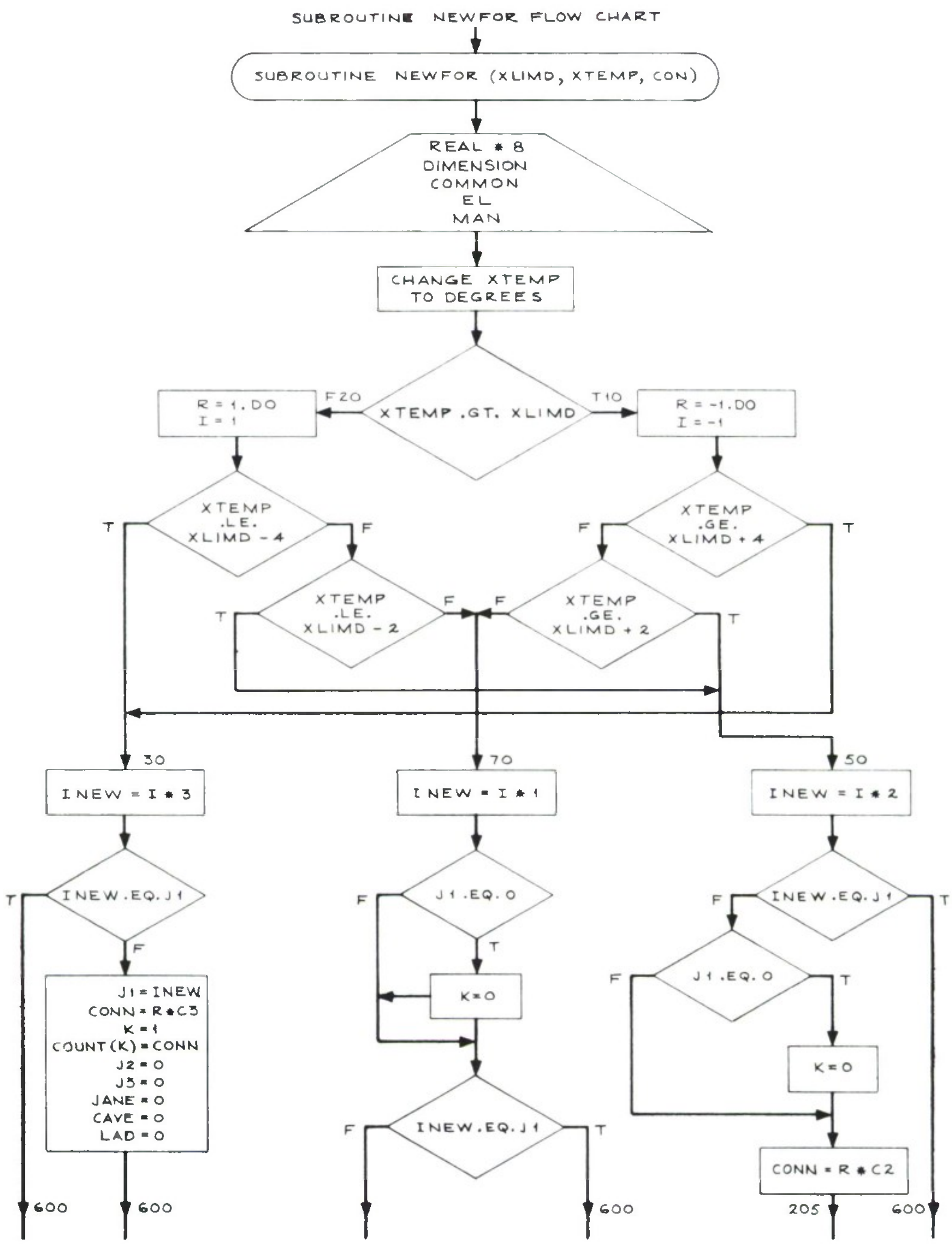
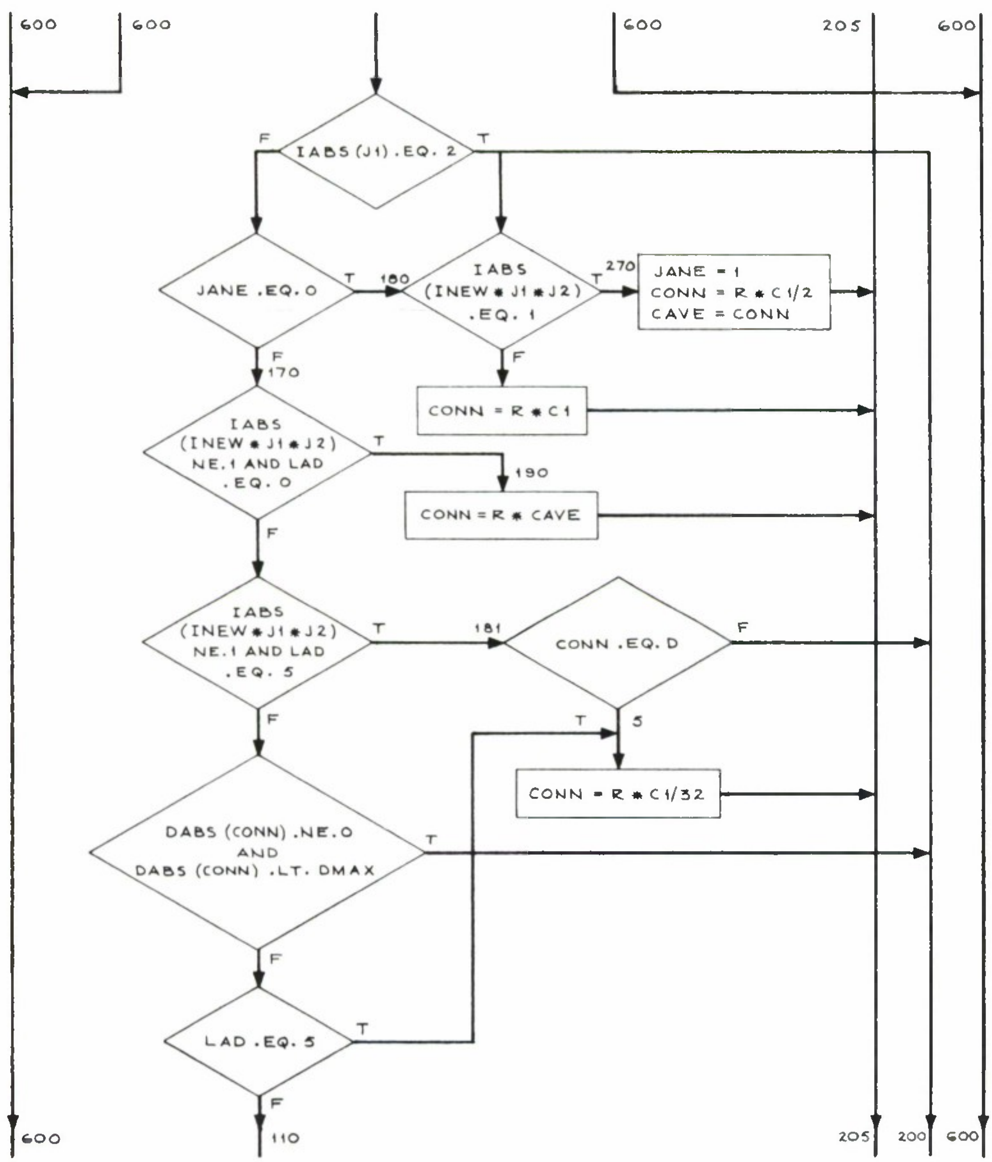


Fig. 19. Example of possible use of force scheme of A. Braga-Illa.

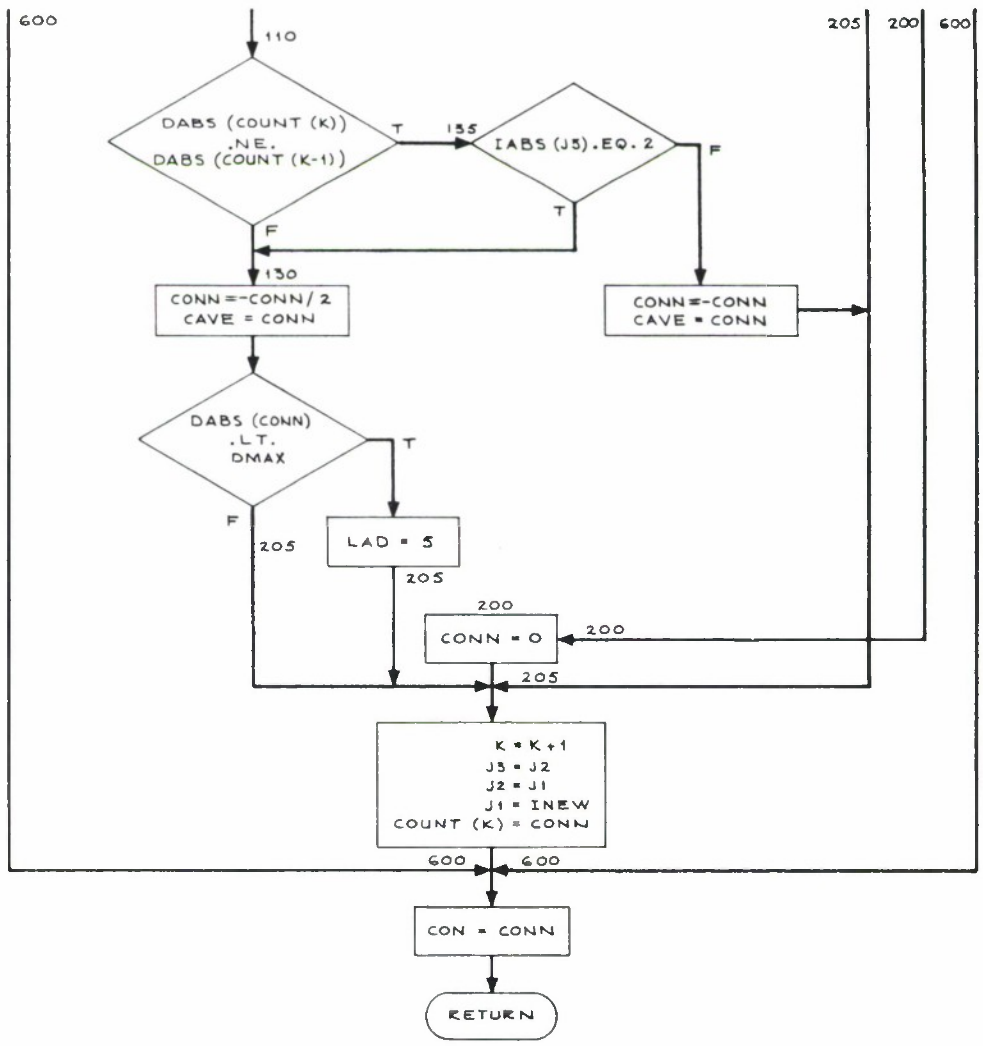
3-63-8498


SUBROUTINE NEWFOR FLOW CHART
SUBROUTINE NEWFOR (XLIMD, XTEMP, CON)
REAL * 8
DIMENSION
COMMON
EL
MAN
CHANGE XTEMP TO DEGREES
XTEMP .GT. XLIMD
F20
T10
R = 1.DO
I = 1
R = -1.DO
I = -1
XTEMP .LE. XLIMD - 4
XTEMP .GE. XLIMD + 4
XTEMP .LE. XLIMD - 2
XTEMP .GE. XLIMD + 2
T
F
30
70
50
INEW = I * 3
INEW = I * 1
INEW = I * 2
INEW.EQ.J1
J1.EQ.0
K=0
J1 = INEW
CONN = R*C3
K = 1
COUNT(K) = CONN
J2 = 0
J3 = 0
JANE = 0
CAVE = 0
LAD = 0
INEW.EQ.J1
J1.EQ.0
K=0
CONN = R * C2
600
600
600
205
600

3-63-8498


600
600
600
205
600
IABS (J1) .EQ. 2
F
T
JANE .EQ. 0
T
180
IABS (INEW * J1 * J2) .EQ. 1
T
270
JANE = 1
CONN = R * C1/2
CAVE = CONN
F
170
F
CONN = R * C1
IABS (INEW * J1 * J2) NE. 1 AND LAD .EQ. 0
T
190
CONN = R * CAVE
F
IABS (INEW * J1 * J2) NE. 1 AND LAD .EQ. 5
T
181
CONN .EQ. D
F
F
T
5
CONN = R * C1/32
DABS (CONN) .NE. 0 AND DABS (CONN) .LT. DMAX
T
F
LAD .EQ. 5
T
F
600
110
205
200
600

3-63-8498



600
110
205
200
600
DABS (COUNT (K))
.NE.
DABS (COUNT (K-1))
T
135
IABS (J3).EQ. 2
F
T
F
130
CONN =-CONN/2
CAVE = CONN
CONN =-CONN
CAVE = CONN
DABS (CONN)
.LT.
DMAX
T
F
205
LAD = 5
205
200
CONN = 0
200
205
K = K+1
J3 = J2
J2 = J1
J1 = INEW
COUNT (K) = CONN
600
600
CON = CONN
RETURN

APPENDIX F

DOUBLE PRECISION FUNCTION CORR (TEL) returns to FLAM an angle by which the longitude XLAM is thereby modified. In FLAM: X = CORR(TEL), XTEMP = XLAM - X. The program calculates the apparent right ascension of the sun and the time of ephemeris transit, good for the year 1967, and accurate to about four seconds of arc of the right ascension and about 1.5 seconds on the ephemeris transit. There are eight options in this program determined by the 13 variable. Two of the options call the SUBROUTINE CURT, and two of the options call the SUBROUTINE ALVISE. When I3 = 4, the logic in FLAM bypasses the call to CORR.

If I3 is 7 or 8, immediately after converting TEL from seconds of days, the program branches to 110. Day is modulated by 365, and CURT is called to determine VALUE. If I3 = 8, VALUE is changed to radians and a random error is added on. If I3 = 7, CORR is simply VALUE in radians and control is returned to FLAM. CURT picks up an ephemeris sun sensor value from the step function tables stored in the BLOCK DATA/SPETS/. These tables simulate the logic in an electronic sun sensor within the satellite.

For the other options, the program calculates the mean anomaly, sun's coordinates, referenced to a geocentric coordinate system, right ascension of the sun, right ascension of the mean sun, and DIFF the difference between the right ascension of the sun and the right ascension of the mean sun added to 6.28 when |DIFF| is less than 3. Now the program branches according to I3 to calculate VALUE. In each case after VALUE is calculated, control goes to statement 31 where CORR = DIFF - VALUE.

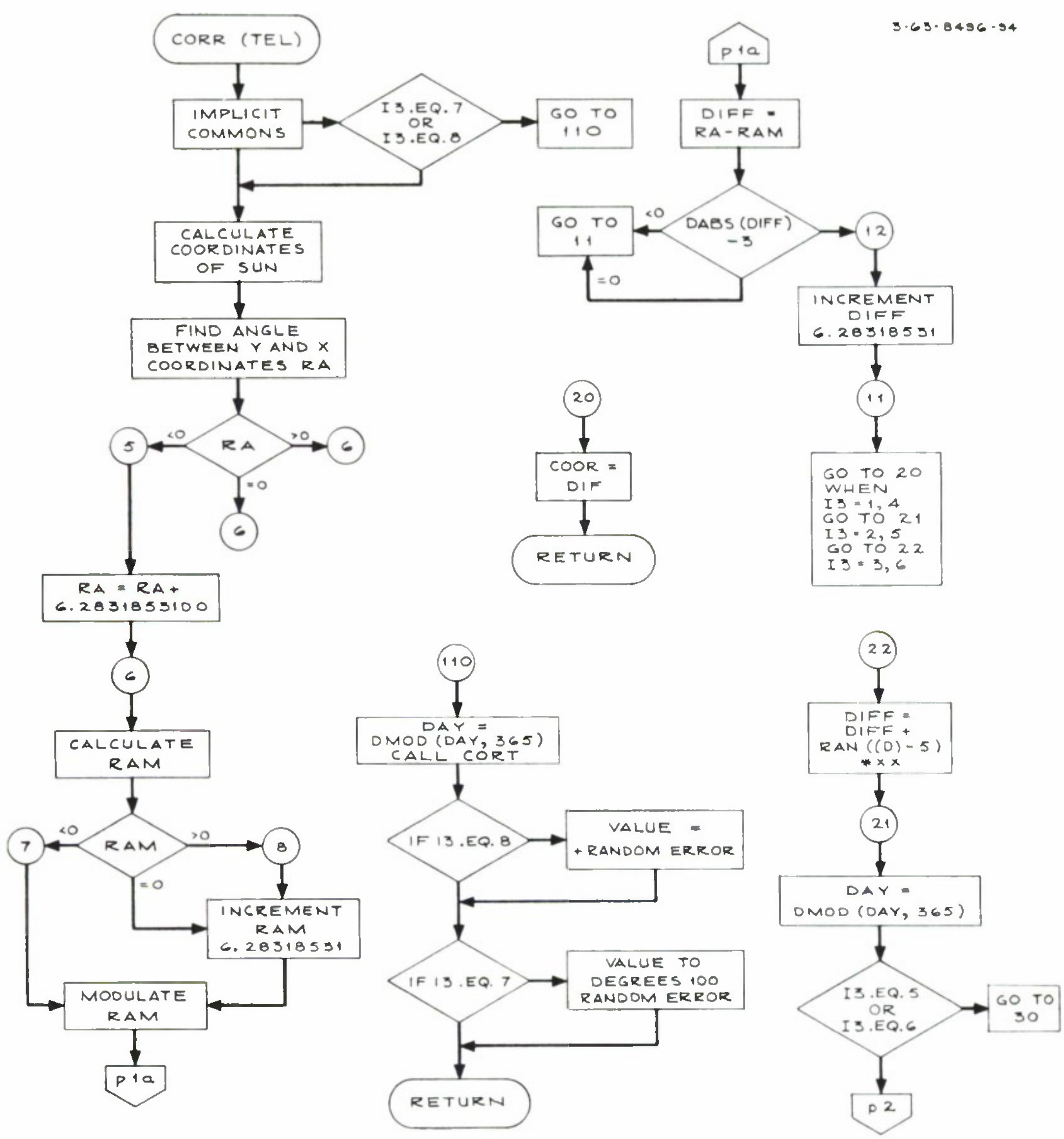
If I3 is 5 or 6, VALUE is returned from a call to SUBROUTINE ALVISE. When I3 = 6 DIFF has a random error added in. ALVISE as a subroutine operates identical to CURT, looking up a value in a BLOCK DATA. When control is returned to CORR, CORR branches to 31.

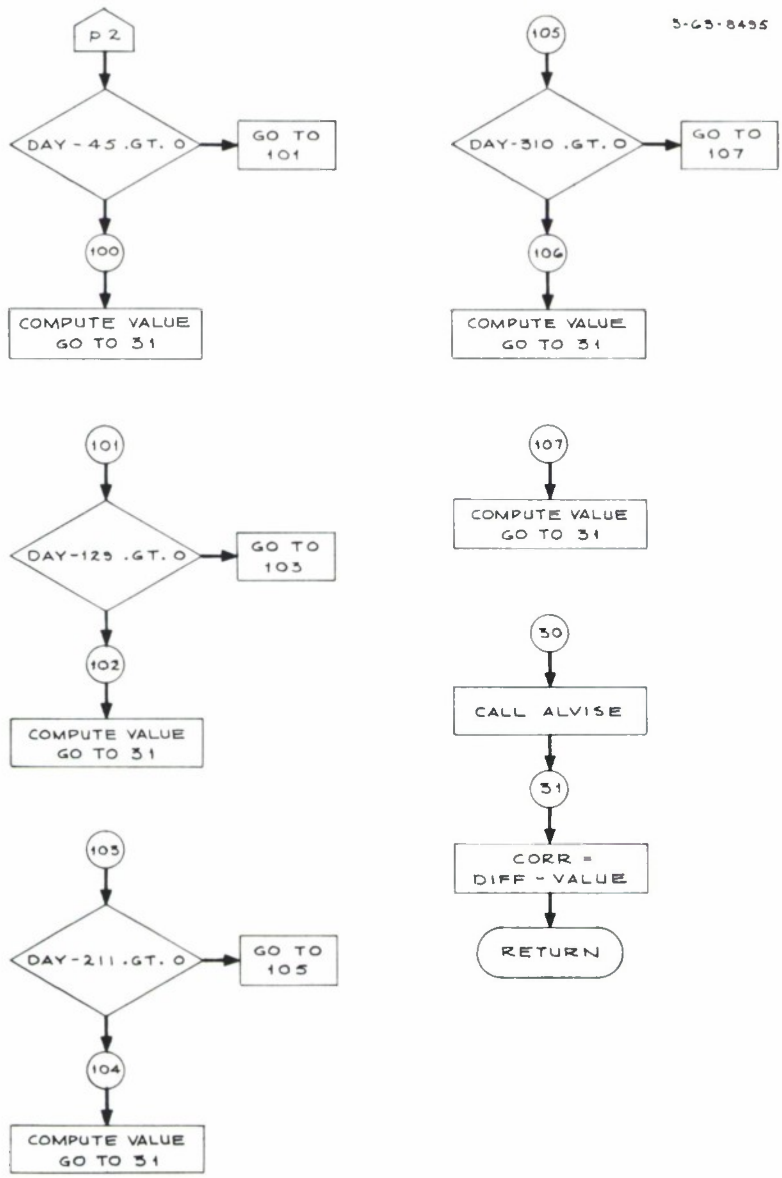
If I3 = 1, CORR = DIFF and control is returned to FLAM. When I3 = 2, DAY is modulated by 365. The formula for VALUE is determined by which part of the year DAY falls.

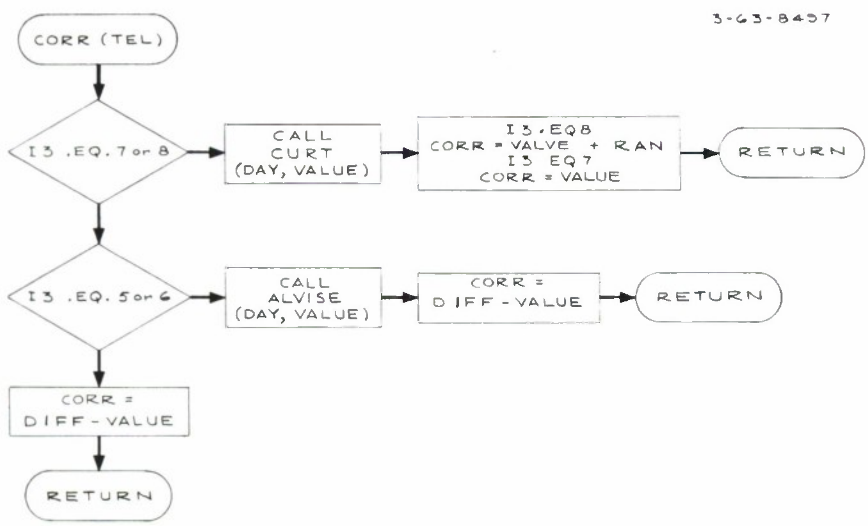
Day	Value
0 - 45	100 VALUE = .08D0 * (DAY + 2.25D0) * XX
46 - 129	102 VALUE =-.0584D0 * (DAY - 111.D0) * XX
130 - 211	104 VALUE = .033D0 * (DAY - 162.D0) * XX
212 - 310	106 VALUE =-0.0584D0 * (DAY - 240.D0) * XX
310 - 360	107 VALUE = .08D0 * (DAY - 363.D0) * XX

After each calculation of VALUE control branches to 31. When I3 is 3, DIFF is modified using the random error function, then logic follows the same for I3 = 2. There is a loop for I3 = 4, although CORR is not called when I3 = 4, to avoid an error message from the compiler.

FUNCTION CORR FLOW CHART




3-63-8495
P2
DAY - 45 .GT. 0
GO TO 101
100
COMPUTE VALUE GO TO 31
101
DAY-125 .GT. 0
GO TO 103
102
COMPUTE VALUE GO TO 31
103
DAY-211 .GT. 0
GO TO 105
104
COMPUTE VALUE GO TO 31
105
DAY-310 .GT. 0
GO TO 107
106
COMPUTE VALUE GO TO 31
107
COMPUTE VALUE GO TO 31
30
CALL ALVISE
31
CORR = DIFF - VALUE
RETURN


3-63-8457
CORR (TEL)
I3 .EQ. 7 or 8
CALL
CURT
(DAY, VALUE)
I3 . EQ8
CORR = VALVE + RAN
I3 EQ7
CORR = VALUE
RETURN
I3 .EQ. 5 or 6
CALL
ALVISE
(DAY, VALUE)
CORR =
DIFF - VALUE
RETURN
CORR =
DIFF - VALUE
RETURN

APPENDIX G

The SUBROUTINE TACK (XLAM) and its ENTRY CHAN (XLAM) return to MAIN via FLAM three forces, FORCCT, FORLAM, and ALONGR, which are used to calculate D2THET, D2XLAM, and D2R respectively. The ENTRY CHAN(XLAM) fixes the sail along the radius to the satellite from the earth. According to the I8 variables, the sail remains upright on the radius or is flipped throughout part of the orbit. FORCCT is the solar force exerted by the sail in the direction of theta. FORLAM is the solar force exerted by the sail in the λ direction. ALONGR is the solar radiation force exerted along the radius. L. Black has pointed out that the most efficient sail would be one of high thermal emissivity on both sides with a high solar absorptivity on one side and a high solar reflectivity on the other. A sail made out of aluminized microsheet stuck on an aluminum sheet painted black on the back side would be a good approximation to this. The average force along the orbit for a sail whose plane is coincident with the orbit radius vector and the north-south direction is given approximately by

$$F = \frac{AS}{2C}$$

where

A = is the area of the sail

S = solar constant

C = velocity of light

The sail placed on angle bisector of angle formed by sun and orbital tangent in direction of the motion is the most efficient sail; the program allows also an offset from this position, OFFSET.

When the solar rays hit the sail they generate two forces which must be resolved into the three components needed. The two forces are one a force perpendicular to the sail itself and opposite to the side of the sail hit by the solar radiation; the second is a force along the sail due to its various

absorptivity properties. Both sides of the sail are reflective when the sail is "tacked" and when it is flipped. When ENTRY CHAN(XLAM) is used, the sail is fixed on the radius from the earth to the satellite, and one side is highly reflective while the other side is highly absorptive. The three components of the generated forces are the force along the orbit, FORLAM; the force along the radius vector of the satellite, ALONGR, and the force along λ, FORLAM.

In this subroutine all the vectors are unit vectors and lie in the orbital plane of the satellite. The coordinate system is geocentric, with the X-axis lying along Aries. XNORMA, YNORMA are the coordinates of the normal to the sail itself. The coordinates of the sun are calculated as in Moulton pp 171, and then the vector is projected into the orbital plane with coordinates XPROJ, YPROJ. XFORCE, YFORCE are the coordinates of the orbital tangent. XSAIL, YSAIL are the coordinates of the normal to the sail. XRESAL, YRESAL are the coordinates of the sail itself. The angle between the sail and the radius is ANCHOR. DOTPRO is the cosine of the angle between the normal to the sail and the projection of the sun. The absolute value of the cosines is used to simplify the logic in determining quadrants. ROCK is the absolute value of the cosine of the angle between the sail and the radius. TRAP is the absolute value of the cosine of the angle between the sun and the normal to the sail. When DOTPRO is less than zero the sail is receiving the solar force on its absorptive side. POT is the force exerted perpendicular to the sail due to the solar radiation pressure. POT = (2 - A) * DOTPRO ** 2. ALOSAL is the force along the sail due to a non-perfect sail.

ALOSAL = -A * TRAP * (XSUN * SRESAL + YSUN * YRESAL)

There is a logical sequence to test whether the sail is "ON" with the flipped sail logic. If it is on one continues, if not the program branches to 40, where a sequence of equations set up the small forces when the sail is turned "OFF". Four variables are used to set up the signs of the forces according to the quadrant the sail, sun, and orbital tangent are in. They are COD, FISH, HERRING, and SALT. They are initially +1 and are made -1 when necessary. The solar

force is exerted opposite to the normal to the sail when the angle between the sun's projection and the sail is between zero and 180; the solar force is exerted along the normal to the sail when the angle between the sun's projection and the sail is between 180° and 360°. The coordinates of the solar force are XSORAF, YSORAF. HERRING is negative when the angle between the solar force and the orbital tangent is greater than 180. SALT is negative when the angle between the sail and the orbital tangent is greater than 180. COD is negative when the angle between the solar force and the radius is greater than 180. FISH is negative when the angle between the radius and the sail is greater than 180. The three components of the solar radiation force acting on the satellite are:

$$\left.\begin{array}{lcl} \mathrm{FORLAM} & = & \mathrm{HERRING} * \mathrm{POT} * \mathrm{ROCK} + \mathrm{SALT} * \mathrm{ALOSAL} * \mathrm{BOAT} \\ \mathrm{ALONGR} & = & \mathrm{COD} * \mathrm{POT} * \mathrm{BOAT} + \mathrm{FISH} * \mathrm{ALOSAL} * \mathrm{ROCK} \\ \mathrm{FORCCT} & = & -\mathrm{AB1} * \mathrm{TRAP} * \mathrm{ZSUN} \end{array}\right\} \text{See Figure 20}$$

When the sail is flipped to its "OFF" position there are slight forces acting upon the satellite. DOTPR1 is the cosine of the angle between the sun (not its projection) and the sail. DOTPR2 is the absolute value of ZSUN. FORCCT is negative when ZSUN is positive. Otherwise the three forces are:

$$\begin{array}{lcl} \mathrm{FORLAM} & = & \mathrm{AB1} * \mathrm{DOTPR2} * \mathrm{DOTPRO} \\ \mathrm{ALONGR} & = & -\mathrm{AB1} * \mathrm{DOTPR2} * \mathrm{DOTPR1} \\ \mathrm{FORCCT} & = & (2 - \mathrm{AB1}) * \mathrm{DOTPR2} \end{array}$$

AB1 is the absorptivity received by the program as A1 and A2 and set equal to the proper one depending on the quadrant one is in. Following is a diagram of the forces for an offset sail in one quadrant.

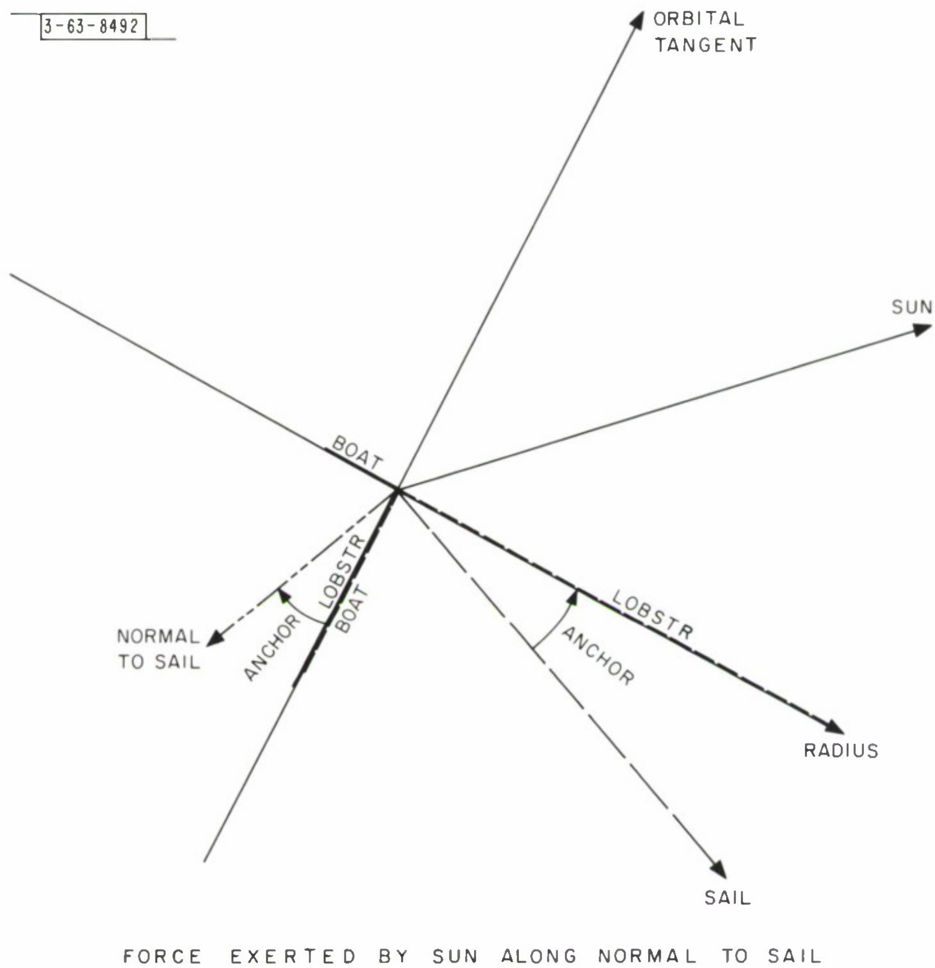
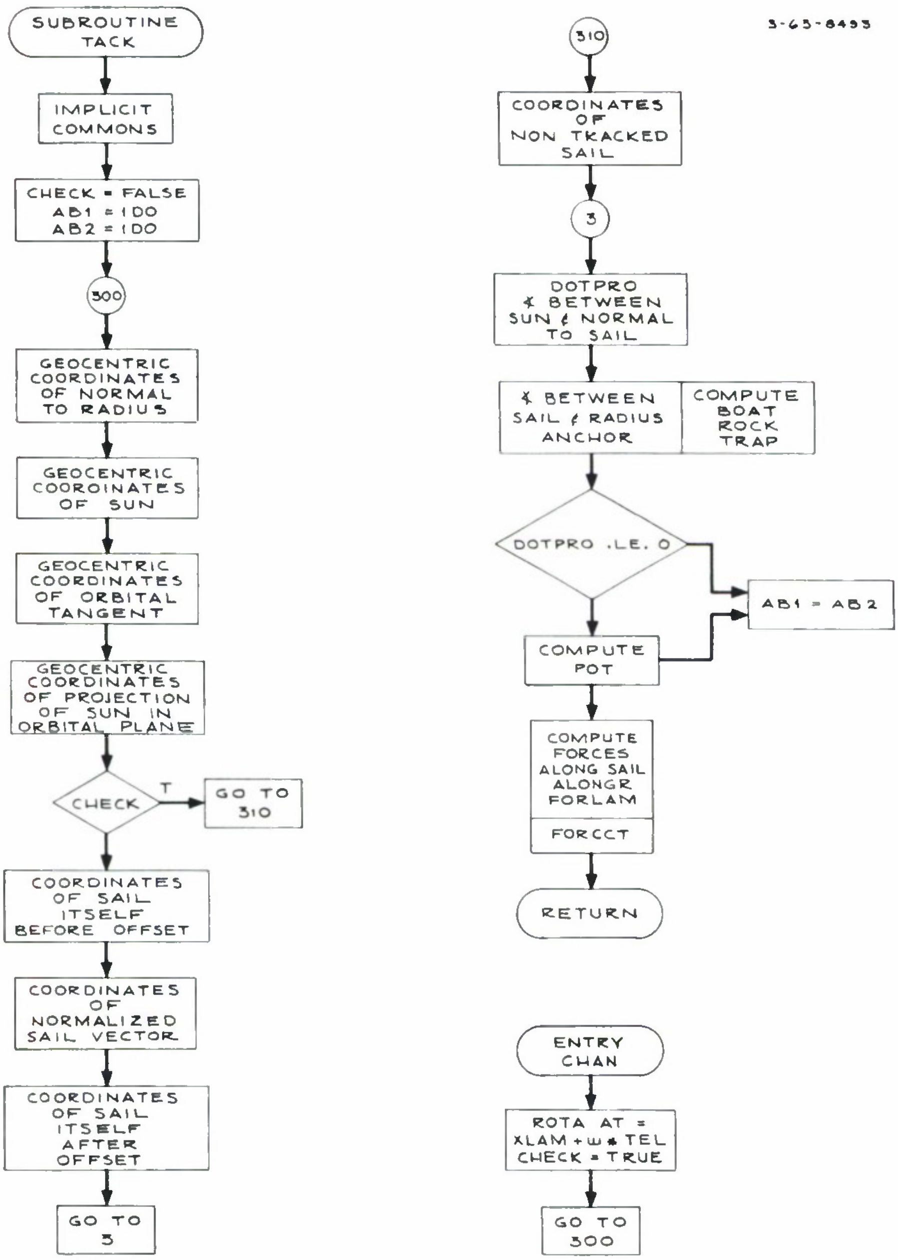


Fig. 20. VECTOR DIAGRAM OF FORCES acting on solar sail.

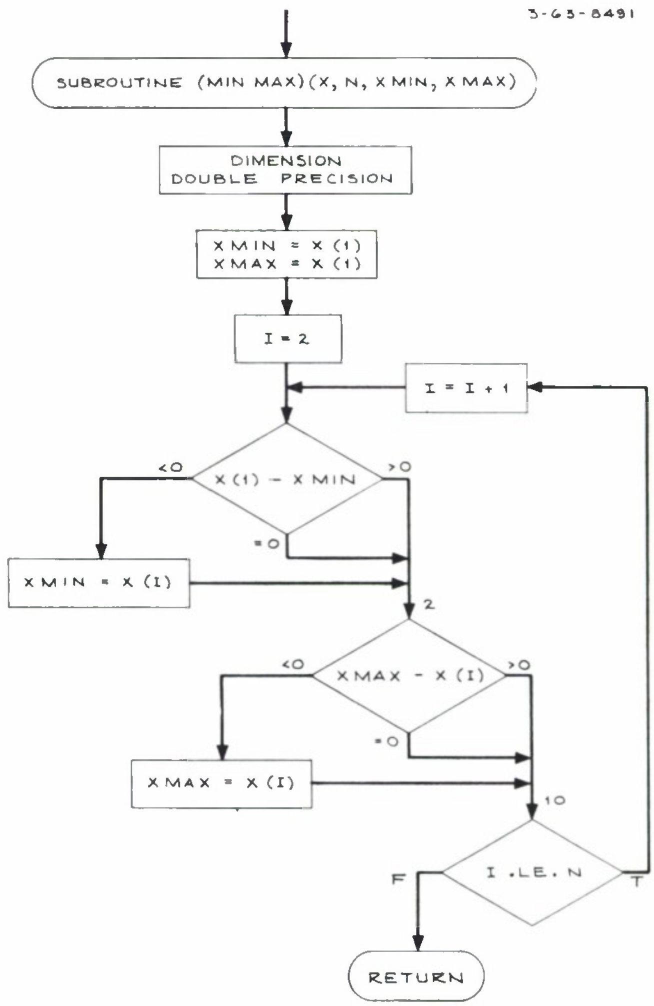
SUBROUTINE TALK FLOW CHART



APPENDIX H

SUBROUTINE MINMAX (X, N, XMIN, XMAX)

Subroutine minmax is designed to find the maximum and minimum value of a set of N numbers.

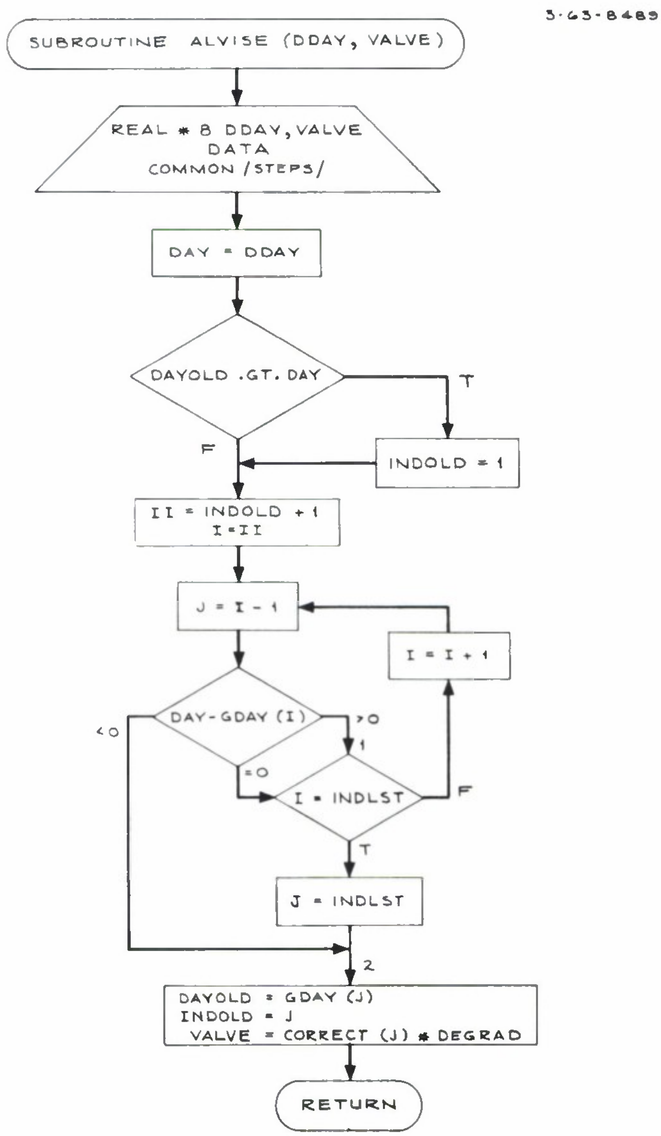

3-63-8491
SUBROUTINE (MIN MAX)(X, N, X MIN, X MAX)
DIMENSION
DOUBLE PRECISION
X MIN = X (1)
X MAX = X (1)
I = 2
I = I + 1
X (1) - X MIN
<0
>0
=0
X MIN = X (I)
2
X MAX - X (I)
<0
>0
=0
X MAX = X (I)
10
I .LE. N
F
T
RETURN

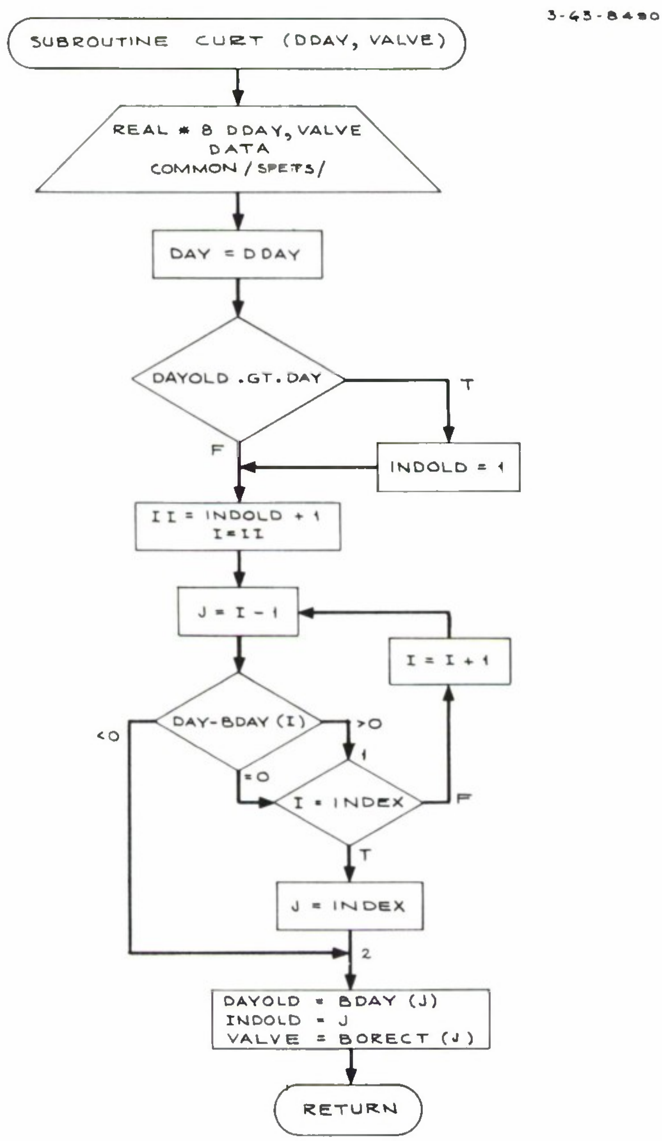
APPENDIX I

SUBROUTINES ALVISE AND CURT

SUBROUTINE ALVISE contains the information for the step function approximation to the equation of time used for the LES-6 satellite.

SUBROUTINE CURT contains the information for the step function approximation to the equation of time used for the LES-6 satellite analemma sensor.


3-63-8489
SUBROUTINE ALVISE (DDAY, VALVE)
REAL * 8 DDAY, VALVE
DATA
COMMON /STEPS/
DAY = DDAY
DAYOLD .GT. DAY
T
F
INDOLD = 1
II = INDOLD + 1
I = II
J = I - 1
I = I + 1
DAY - GDAY (I)
<0
>0
=0
1
I = INDLST
F
T
J = INDLST
2
DAYOLD = GDAY (J)
INDOLD = J
VALVE = CORRECT (J) * DEGRAD
RETURN


3-43-8480
SUBROUTINE CURT (DDAY, VALVE)
REAL * 8 DDAY, VALVE
DATA
COMMON / SPETS /
DAY = DDAY
DAYOLD .GT. DAY
T
F
INDOLD = 1
II = INDOLD + 1
I = II
J = I - 1
I = I + 1
DAY - BDAY (I)
<0
>0
=0
1
I = INDEX
F
T
J = INDEX
2
DAYOLD = BDAY (J)
INDOLD = J
VALVE = BORECT (J)
RETURN

APPENDIX J

LISTING OF THE PROGRAM

COMPILER OPTIONS - NAME= MAIN,OPT=00,LINECNT=50,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID

```
C  THIS VERSION OF Z800 HAS NO REAL I6 OPTION IN IT.
C      XLAM MEASURED EAST OF GREENWICH
C
C         IF I1 = 1, 2ND ORDER HARMONICS USED
C         IF I1 = 2, 3RD ORDER HARMONICS USED
C         IF I2 = 1, PUNCH OUTPUT
C         IF I2 = 2, NO PUNCH OUTPUT
C         I3 CONTROLS SUN CORRECTION     I3 = 1  NO CORRECTION
C                                        I3 = 2 CORRECTED LINEARLY
C                                        I3 = 3 CORRECTED LINEARLY W/ RANDO
C                                        I3 = 4 CORRECTED PERFECTLY
C                                        I3 = 5 CORRECT BY STEP FUNCTION
C                                        I3 = 6 CORRECTED BY S-F W/ RANDOM
C                                     I3 = 7 CORRECT BY EPHEMERIS SUN SENS
C                                        I3 = 8 CORRECT BY EPHEMERIS SS W/R
C                                        I3 = 9 1ST SUN SAIL CORRECTION
C                                               CONSTANT FORCE ALONG LAMDA
C                                        I3 =10 SOLAR SAIL3 FIXED SAIL
C
C         IF I4 = 1 XJ22=-1.816E-6 AND XLAM22 = -15.0*.01745
C         IF I4 = 2 XJ22=-1.700E-6 AND XLAM22 = -19.0*.01745
C         IF I5 = 1, NO FORCE CUTDOWN
C         IF I5 = 2, FORCE CUT BY INPUT NO. CUT
C         I6  =  1  SOLAR SAIL2
C         IF I7 = 1, PLOTTING AVERAGE OF XLAM FOR 10TH DAY
C         IF I7 = 2, PLOTTING XLAM AT END OF EVERY 10TH DAY
C      IF I8 = 3, SAIL IS FLIPPED BY LOGIC IN SATELLITE
C
C  IF NEWFR=TRUE, USE NEW FORCE CALCULATIONS
C  IF NEWFR=FALSE, USE OLD FORCE CALCULATIONS
C
C  CONST IS CALCULATED IN THE FOLLOWING MANNER--CONST=F/M EXPRESSED IN
C  KILONEUTONS PER KILOGRAM.  FOR OUR CASE
C          =1.6*(FACTOR=5.6)/365.25/(86.4*6) = 4.45E-8
C  THE SIGN SHOULD BE NEGATIVE IN THE SECOND AND FOURTH COORDINATES
C
C      Z800 IS A COMBINATION OF VARIOUS ZERO 6 AND ZERO7 PROGRAMS
C         IT USES EITHOR THE Z6 OR Z7 COMPUTATION CONTROLED BY I1
C         TIME, A, + XL PUNCHED OUT EVERY 10TH DAY UNDER CONTROL OF I2
C
C D A N G E R               D A N G E R
C
C      NEVER PUT COMMON /MAN/ IN MAIN PROGRAM
C
C      WRITTEN BY CHAN CROCKER MIT LINCOLN LABORATORY
C      ROOM C-266 LEXINGTON MASS.  TEL. 617-862-5500     EXT. 5307
COMMENT CARD  EPSIL = - 6.* J22
C
C
```

```
                 C
                 C     ALPHABETIZED LIST OF SUBROUTINES
                 C
                 C             ALVISE (DDAY,VALUE)
                 C             CHAN(XLAM)
                 C             CURT(DDAY,VALUE)
                 C             GETIMP
                 C             MINMAX (X,N,XMIN,XMAX)
                 C             NEWFOR (XLIMD,XTEMP,CDN)
                 C             SETOUT (PUNCH)
                 C             SSL
                 C             TACK (XLAM)
                 C
                 C
                 C     ALPHABETIZED LIST OF DOUBLE PRECISION FUNCTIONS
                 C
                 C             CIND (K,A,B)
                 C             CORR (TEL)
                 C             DPNV (Y,DY)
                 C             FINOV (X,H)
                 C             FLAM (XLAM)
                 C
                 C
ISN 0002                   IMPLICIT REAL*8 (A-H,O-Y)
ISN 0003               DIMENSION RT(1500), TH(1500), XLO(1500)
                 C
                 C     WARNING.... VARIABLES IN DIFFERENT COMMONS MUST HAVE DIFFERENT NAM
                 C
ISN 0004               COMMON/ZB/XLIM  , CONST , XLIMX , CUT   , XNUM  , FUEL  ,
                      1         XX    , WS    , COSI  , ESUN  , SINI  , TEL   , JFK   ,
                      2         LLL   , M     , K     , NUM   , I3    , I5
                 C
ISN 0005               COMMON/PARAMS/FITHDY,CONSTI, XLIMXD, XOISP, XMASS , CUTBY , A     ,
                      1              E     ,XI    , OL    , XLAMD , TEND  , ALONGR, DATE,
                      2              FORCCT,FORLAM, A1    , A2    , W     , I1    , I2,
                      3              I3T   ,I4    , I5T   , I6    , I7    , NUMIN
ISN 0006               COMMON/SUPSTF/   AB1,   AB2, OFFSET,    I8
                 C
                 C
ISN 0007                COMMON/DDVE/XLIMD
ISN 0008                COMMON/EL/CAVE,J1,J2,J3,JANE,KAY,LAD
ISN 0009                  EQUIVALENCE (XISP,XDISP)
                 C
ISN 0010               DATA    G,ZAP,GM,DPRINT,DT,RE  /9.8D0,-1.D0,3.986267705,86400.D0,
                      *                                        86.4D0,6378.38800 /
ISN 0011               DATAXJ20,XJ30,XJ31,XJ32,XJ33/1082.48D-6,-2.56D-6,-1.4D-6,
                      I            -.1D2D-6,-.171D-6/
                 C
ISN 0012               XX     = 0.17453292519943290-D1
ISN 0013               W      = .72921151466715650-4
```

```
ISN 0014          ESUN  = 0.0167230D                                                  0103
ISN 0015          WS     = 282.3728700 * XX                                            0104
ISN 0016           SIN1= OSIN (23.44358DO * XX)                                        0105
ISN 0017          CDSI  =OCDS(23.4435800 * XX)                                         0106
ISN D018          XDT = OT                                                             0107
ISN 0019          NDUT  = 6.2831853DD / (W * DT)                                       0108
ISN 0020        7  CALL     GETINP                                                     0109
ISN D021          IF (NUMIN .LT. 0)  GD TO 1D                                          0110
            C                                                                          0111
ISN D023          J1=1
ISN 0024          J2=-1
ISN 0025          J3=1
ISN D026           KAY = D                                                             0115
ISN 0027          JANE=1
ISN 0028          LAD=5
ISN D029          CAVE=CONST/32
            C                                                                          0119
            C      MOVE THE ELEMENTS FROM THE NAMEO COMMON TO BLANK COMMON             0120
ISN DD30           NUM      =   NUMIN                                                  0121
ISN DD31           CDNST    =   CDNSTI                                                 0122
ISN OD32           I3       =   I3T                                                    0123
ISN OD33          I5      =   I5T                                                      0124
ISN DD34           CUT      =   CUT8Y                                                  0125
ISN DD35           TWTYDY   =   FITHDY*DPRINT                                          0126
            C                                                                          0127
ISN 0036          WRITE (6, 9)
ISN D037        9 FURMAT(126H      TOAYS            R              DR            T     0129
                 1            DT             LAM             DLAM             E. LONG  0130
                 2                           )                                         0131
ISN 0038          NYR = TENO                                                           0132
ISN D039             IF(I2 .EQ. 1)  WRITE(7,46) NYR                                    0133
ISN 0D41       46 FDRMAT (15)                                                          0134
            C                                                                          0135
            C                                                                          0136
            C                                                                          0137
            C                                                                          0138
ISN D042          GD TO (27, 28),I4                                                    0139
ISN 0043       27 XJ22  = -1.8160-6                                                    0140
ISN 0044          GO TO 29                                                             0141
            C                                                                          0142
ISN 0045       28 XJ22   = -1.70-6                                                     0143
ISN 0046           XLAM22 = -19.D0 * XX                                                0144
            C                                                                          0145
ISN 0D47       29 CDNTINUE                                                             0146
ISN 0048          XLAM31 = -168.00 * XX                                                0147
ISN 0049          XLAM32= D.D0                                                         0148
ISN 0050          XLAM33 = 24.9 * XX                                                   0149
ISN D051          EPSIL = -6.D0 * XJ22                                                 0150
ISN 0D52          XJ2 = XJ20                                                           0151
            C                                                                          0152
```

```
ISN 0053          C3 = GM * EPSIL * RE * RE
ISN 0054          C1     =(C3/EPSIL) * XJ2 * 3.00/2.00
ISN 0055          C2     = C3 * 3.00/2.00
ISN 0056          C4     = C3 / 2.D0
            C
ISN 0057           M=0
ISN 0058          K=-1
ISN 0059          JFK=0
ISN 0060          N = 0
ISN 0061          XNUM = NUM
ISN 0062          XNUM  = XNUM * 1000.00 * XMASS * XOT / (XISP * G)
ISN 0063          FUEL  = 0.00
ISN 0064           XLAM22 = -15.400 *XX
ISN 0065          XLIM = XLIMO*XX
ISN 0066          XLIMX = XLIMXO*XX
ISN 0067          T      = OATE  * 86400.00
ISN 0068          TPRINT= T + 86400.00
ISN 0069          TOUT = TWTYOY + T
ISN 0070          V      = (90.D0-OL)*XX
ISN 0071          P      = A * (1.00 - E*E)
ISN 0072          R      = P / (1.00 + E * OCOS(V) )
ISN 0073          H      = OSQRT(GM*P)
ISN 0074          OR     = H * E * OSIN(V)/              P
ISN 0075          THETA = 90.00 - XI
ISN 0076          THETA = THETA*XX
ISN 0077          TENO  = TENO * 86400.00
ISN 0078          XLOEY = XLAMO
ISN 0079          XLAM  =  XLAMO * XX
ISN 0080          OFFSET = OFFSET * XX
ISN 0081          OTHETA= 0.00
            C
            C     CALCULATE OXLAM
ISN 0082          O2R   = H*H*(1.00-R/P)/R**3
            C     BETTER CALCULATION OF OXLAM  D2R IS NOT 0
            C
ISN 0083          GO TO (2, 52),I1
            C
ISN 0084        2  SQ  = GM/(R**3 * (OSIN(THETA))**2)
                 1       -3.00*GM*XJ2*RE*RE*
                 2      (3.00*(OCOS(THETA))**2-1.00)/(2.00*R**5*(DSIN(THETA))**2)
                 3       +3.00 * C3           *DCOS(2.00*(XLAM-XLAM22)) / (2.D0*R**5)
                 4      + D2R / (R *(DSIN(THETA))**2)
            C
ISN 0085          DXLAM = DSQRT(SQ)-W
            C
            C
ISN 0086           CALL SSL
            C
ISN 0087        1 SITH  = OSIN(THETA)
ISN 0088          SITHSO= SITH * SITH
```

```
ISN 0089          COTH   = DCOS(THETA)
ISN 0090          COTHSQ= COTH * COTH
ISN 0091          SIXLT2= DSIN(2.DO * (XLAM-XLAM22))
ISN 0092          COXLT2= DCOS(2.DO * (XLAM-XLAM22))
ISN 0093          SITHT2 = DSIN (2.DO * THETA)
ISN 0094          DXLW   = DXLAM + W
ISN 0095          DXLWSQ= DXLW*DXLW
ISN 0096          R4     = R**4
ISN 0097          R5     = R4*R
ISN 0098          TEL = T
ISN 0099          XFORCE = FLAM (XLAM)
               CC
ISN 0100          D2R    = R * DTHETA**2 + R * SITHSQ * DXLWSQ
                 1       -GM/R**2 + C1 * (3.DO * COTHSQ -1.DO) / R4
                 2                                 - C2 * SITHSQ * COXLT2      / R4
                 3       + XFORCE * ALONGR
               CC
ISN 0101          D2XLAM=-C3 * SIXLT2 / R5
                 1       -DXLW * COTH * 2.DO * DTHETA / SITH
                 2       -2.DO * DR * DXLW /R
                 3       + (XFORCE / (SITH * R)) * FORLAM
               CC
ISN 0102          D2THET=-2.DO * DR * DTHETA /R
                 1       +DXLWSQ * SITHT2 /2.DO
                 2       + C1 * SITHT2 / R5
                 3       + C4 * SITHT2 * COXLT2 / R5 + XFORCE * FORCCT / R
ISN 0103          GO TO 23
               C
ISN 0104       52 SITH   = DSIN(THETA)
ISN 0105          SITHSQ= SITH * SITH
ISN 0106          COTH   = DCOS(THETA)
ISN 0107          COTHSQ= COTH* COTH
ISN 0108          REDR   = RE/R
ISN 0109          REDRSQ= REDRSQ
               C
ISN 0110          FR     = GM/(R**2) * (-1.DO + REDRSQ *
                 1       (1.5 * XJ20 * (3.DO * COTHSQ - 1.DO )
                 2       + 9.DO * XJ22 * SITHSQ * DCOS (2.DO*(XLAM-XLAM22))
                 3       + 2.DO * REDR * XJ30 * (5.DO * COTHSQ -3.DO)* COTH
                 4       + 6.DO * REDR * XJ31 * (5.DO * COTHSQ -1.DO)* SITH *
                 5        DCOS(XLAM-XLAM31)
                 6       +60.DO * REDR * XJ32 * SITHSQ*COTH*DCOS(2.DO*(XLAM-XLAM32))
                 7       +60.DO * REDR * XJ33 * SITH*SITHSQ* DCOS(3.DO*
                 8            (XLAM-XLAM33))      ))
               C
               C      BETTER CALCULATION OF DXLAM  D2R IS NOT 0
ISN 0111          SQ     = (D2R -FR) / (R * SITHSQ)
ISN 0112          DXLAM = DSQRT(SQ) - W
               C
               C
```

```
ISN 0113          CALL SSL
               C
ISN 0114       51 SITH  = OSIN (THETA)
ISN 0115          SITHSQ= SITH * SITH
ISN 0116          SITHCU= SITHSQ * SITH
ISN 0117          SITHT2= OSIN (2.00 * THETA)
ISN 0118          COTH  = OCOS (THETA)
ISN 0119          COTHSQ= COTH * COTH
ISN 0120          COTHS5= 5.DO * COTHSQ
ISN 0121          SICOTH= SITH*COTH
ISN 0122          R2    = R* R
ISN 0123          R4    = R**4
ISN 0124          R5    = R4*R
ISN 0125          REDR  = RE/R
ISN 0126          REORSQ= REOR*REOR
ISN 0127          REXJ30= REDR * XJ30
ISN 0128          REXJ31= REDR * XJ31
ISN 0129          RXCS51= REXJ31 * (COTHS5 -1.DO)
ISN 0130          REXJ32= REOR * XJ32
ISN 0131          REXJ33= REDR * XJ33
ISN 0132          RXSISQ= REXJ33 * SITHSQ
ISN 0133          GMDRSQ= GM/R2
ISN 0134          ORT2DR= 2.00 * OR/R
ISN 0135          OXLW  = OXLAM + W
ISN 0136          DXLWSQ= OXLW*DXLW
ISN 0137          XLAMT2= 2.DO * XLAM
ISN 0138          SIXLT2= DSIN(XLAMT2)
ISN 0139          COXLT2= OCOS(XLAMT2)
ISN 0140          XL22T2= 2.00 * (XLAM-XLAM22)
ISN 0141          CX22T2= DCOS (XL22T2)
ISN 0142          CXJ22T= XJ22 * CX22T2
ISN 0143          XL31  = XLAM-XLAM31
ISN 0144          COXL31= OCOS (XL31)
ISN 0145          XL32T2= 2.00 * (XLAM-XLAM32)
ISN 0146          CX32T2= OCOS (XL32T2)
ISN 0147          XL33T3= 3.DO * (XLAM-XLAM33)
ISN 0148          TEL = T
ISN 0149          XFORCE = FLAM (XLAM)
               C
ISN 0150          FR    = GMDRSQ * (1.DO
                 1         +REDRSQ * (1.5DO * XJ20 * (3.00 * COTHSQ -1.DO )
                 2         + 9.DO * SITHSQ * CXJ22T
                 3         + 2.DO * REXJ30 * (COTHS5 -3.DO) * COTH
                 4         + 6.DO * RXCS51 * SITH * COXL31
                 5         + 60.DO * REXJ32 * SITHSQ * COTH * CX32T2
                 6         + 6C.DO * REXJ33 * SITHCU * OCOS (XL33T3)    ))
                 7         + XFORCE * ALONGR
               C
ISN 0151          FXLAM =  GMDRSQ * REORSQ * (6.DO * XJ22 * SITH * SIX_T2
                 1         +1.5DO * RXCS51 *                       DSIN(XL31)
```

```
                 2        +30.00 * REXJ32              * SICOTH * DSIN(XL32T2)
                 3        +45.00 * RXSISQ                       * DSIN(XL33T3) )
                 4         + XFORCE * FORLAM
               C
ISN 0152            FTHETA= -GMORSQ * REORSQ * (-3.00 * XJ20 * SICOTH
                 1        +6.00 * XJ22 * SICOTH * CX22T2
                 2        -1.500 * REXJ30 * (COTHS5-1.00) * SITH
                 3        +1.500 * REXJ31 * (15.00 * COTHSQ - 11.DO) * COTH * COXL31
                 4        +15.00 * REXJ32 * ( 3.00 * COTHSQ -  1.DO) * SITH * CX32T2
                 5        +45.00 * RXSISQ          * COTH       * DCOS(XL33T3) )
                 6         + XFORCE * FORCCT
               C
ISN 0153            D2R    = R * OTHETA**2 + R * SITHSQ * OXLWSQ + FR
               C
ISN 0154            O2XLAM= -2.00 * OR * OXLW/R
                 1        -2.00 * COTH * OTHETA * OXLW/SITH + FXLAM/(R*SITH)
               C
               C
ISN 0155            D2THET= -2.DO * DR * OTHETA/R + FTHETA/R + DXLWSQ * SITHT2/2.00
               C
               CHANGE THE FOLLOWING CAROS IF  A OIFFERENT OUTPUT MODE IS DESIRED
ISN 0156          23 IF (T - TOUT) 21,14, 14
ISN 0157          14 IF ( N - NOUT) 15, 15, 16
ISN 0158          15 N = N+ 1
ISN 0159             RT(N) = R
ISN 0160             TH(N) = THETA
ISN 0161             XLD(N) = XLAM
ISN 0162             GO TU 21
ISN 0163          16 CALL MINMAX (RT,N,RP,RA)
ISN 0164             A      = (RA +RP) /2.DO
ISN 0165             E      =  RA/A - 1.DO
ISN 0166             CALL MINMAX (TH, N, TMIN, TMAX)
ISN 0167             XI  = 90.DO - TMIN/XX
ISN 0168             CALL MINMAX (XLD,N,XLMIN,XLMAX)
ISN 0169             XLMID = XLMIN/XX
ISN 0170             XLMAD = XLMAX/XX
ISN 0171             DIFFER = XLMAD - XLMID
ISN 0172             WRITE (6      ,25)A,E,XI,XLMID,XLMAD,DIFFER
ISN 0173          25 FORMAT(4H A = ,1PE15.6,4H E = ,1PE15.6,5H XI =,1PE15.6,5H LMI=
                 11PE15.6,5H LMA=   ,1PE15.6, 5H DIF=   , 1PE15.6)
               C
ISN 0174             GO TO (12,13),I2
ISN 0175          12 GO TO (137,138),I7
ISN 0176         137 Y2     = (XLMID +XLMAD) / 2.00
ISN 0177             GO TO 139
ISN 0178         138 Y2 = XLAM/XX
ISN 0179         139 CONTINUE
ISN 0180             WRITE (7,114) TDAYS, A, Y2
ISN 0181         114 FORMAT (3F20.7)
               C
```

```
ISN 0182         13 TOUT = TOUT + TWTYDY
ISN 0183            N = 0
ISN 0184            GO TO 23
ISN 0185         21 IF(T - TPRINT) 4,3,3
           C
ISN 0186          3 TOEG = THETA/XX
ISN 0187             XLOEG=XLAM/XX
ISN 0188            TDAYS = T/86400.00
ISN 0189            DXLDEG= 86400.DO * DXLAM/XX
ISN 0190            OTDEG = 86400.DO * DTHETA/XX
ISN 0191            XLDET = XLOEG
ISN 0192            IF (XLOET-XLOEY) 34, 34, 35
ISN 0193         34 LLL = -1
ISN 0194            GO TO 36
ISN 0195         35 LLL =  1
ISN 0196         36 XLOEY = XLDEG
ISN 0197            XELONG= 360.DO - XLOEG
ISN 0198             XELONG = 360. -  XLOEG
           C
ISN 0199            WRITE (6     ,5)TDAYS,R,DR,TDEG,DTDEG,XLDEG,DXLDEG ,XELONG
ISN 0200          5 FORMAT(1X,1PRE15.6)
           C
ISN 0201             K=-1
ISN 0202            TPRINT = TPRINT + OPRINT
ISN 0203          4 IF (T.LT.TEND) GO TO 6
ISN 0205            IF (I2.EQ.1) WRITE(7,88) ZAP
ISN 0207         88 FORMAT (E20.5)
ISN 0208            GO TO 7
           C
           C
ISN 0209          6 T=FINDV(T,OT)
ISN 0210            R = DPNV(R,DR)
ISN 0211            THETA = DPNV(THETA,OTHETA)
ISN 0212            XLAM = DPNV(XLAM,DXLAM)
ISN 0213            OR = DPNV(DR,D2R)
ISN 0214            OTHETA = DPNV(DTHETA,O2THET)
ISN 0215            DXLAM = OPNV(DXLAM,O2XLAM)
           C
           C
           C          THE PROGRAM CAN NEVER GO TO S.N. 10  THIS IS HERE TO PREVENT
           C                 A DIAGNOSTIC (PROGRAM DOES NOT END WITH A STOP
ISN 0216               IF(I1 -1) 10,1,51
ISN 0217         10 RETURN
ISN 0218            ENO
```

```
LEVEL 2 FEB 67                              OS/360  FORTRAN H                                        DATE  68.067/17.15.57

          COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=50,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID

   ISN 0002            SUBROUTINE ALVISE(DDAY,VALUE)                                               0397
                 C     SUBROUTINE TO PICK UP A NEW EPHEMERIS CORRECTION VALUE                      0398
                 C     THE ROUTINE ASSUMES TABLES DEFINING A STEP FUNCTION HAVE BEEN CO           0399
                 C             MPILED AND STORED IN   /STEPS/ COMMON                               0400
                 C                                                                                 0401
                 C     THE ROUTINE SCANS THE GDAY ARRAY LOOKING FOR THE FIRST DAY LESS            0402
                 C     THAN THE INPUT DAY.  IT THEN USES THE INDEX FOUND BY THE SCAN TO            0403
                 C     REFERENCE THE CORECT TABLE                                                  0404
                 C                                                                                 0405
   ISN 0003           REAL*8 DDAY,VALUE                                                            0406
   ISN 0004            DATA DAYOLD,INDOLD,DEGRAD/366.,81,.174532925E-1/                            0407
                 C                                                                                 0408
                 C     IN AN ATTEMPT TO SPEED UP THE TABLE LOOK UP   WE ARE KEEPING TRA            0409
                 C     OF THE OLD VALUE ON THE LAST CYCLE AND STARTING THE TLU FROM THE            0410
                 C             PREVIOUS DAY VALUE                                                  0411
   ISN 0005            COMMON /STEPS/ GDAY(200),CORECT(200),INDLST                                 0412
   ISN 0006            DAY = DDAY                                                                  0413
   ISN 0007            IF(DAYOLD .GT. DAY)    INDOLD = 1                                           0414
   ISN 0009            II       =    INDOLD + 1                                                    0415
   ISN 0010            DO I I=II,INDLST                                                            0416
   ISN 0011            J        = I - 1                                                            0417
   ISN 0012            IF(DAY - GDAY(I))      2,I,1                                                0418
   ISN 0013          I CONTINUE                                                                    0419
   ISN 0014             J       =     INDLST                                                       0420
                 C                                                                                 0421
                 C     UNABLE TO FIND A CORREVT VALUE                                              0422
                 C                         IMPOSSIBLE -------------                                0423
                 C                                                                                 0424
   ISN 0015          2 CONTINUE                                                                    0425
   ISN 0016            DAYOLD  =    GDAY(J)                                                        0426
   ISN 0017            INDOLD  =    J                                                              0427
   ISN 0018            VALUE = CORECT(J)* DEGRAD                                                   0428
   ISN 0019            RETURN                                                                      0429
   ISN 0020            END                                                                         0430
```

```
       COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=50,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID
ISN 0002             BLOCK DATA
             C       DATA SUBROUTINE TO DEFINE THE STEP FUNCTIONS FOR AN EPHEMERIS CO
             C
ISN 0003             COMMON /STEPS/ GOAY(200),CORECT(200),INDLST
             C
             C
             C
             C * * * * * * * * * * * * * * * * * * * * * *  * * * * * * * * * * * * *
             C
             C                 STATE O
             C
ISN 0004             DATA GOAY( 1),CORECT( 1) /0.,.25                    /
ISN 0005             DATA GDAY( 2),CORECT( 2) /3.,.50                    /
ISN 0006             DATA GDAY( 3),CORECT( 3) /6.,.75                    /
ISN 0007             DATA GDAY( 4),CORECT( 4) /9.,1.                     /
ISN 0008             DATA GOAY( 5),CORECT( 5) /12.,1.25                  /
ISN 0009             DATA GOAY( 6),CORECT( 6) /15.,1.50                  /
ISN 0010             DATA GOAY( 7),CORECT( 7) /18.,1.75                  /
ISN 0011             DATA GDAY( 8),CORECT( 8) /21.,2.00                  /
ISN 0012             DATA GDAY( 9),CORECT( 9) /24.,2.25                  /
ISN 0013             DATA GDAY(10),CORECT(10) /27.,2.50                  /
ISN 0014             DATA GDAY(11),CORECT(11) /30.,2.75                  /
ISN 0015             DATA GOAY(12),CORECT(12) /33.,3.00                  /
ISN 0016             DATA GDAY(13),CORECT(13) /36.,3.25                  /
ISN 0017             DATA GDAY(14),CORECT(14) /39.,3.50                  /
ISN 0018             DATA GDAY(15),CORECT(15) /42.,3.75                  /
             C
             C
             C
             C * * * * * * * * * * * * * * * * * * * * * *  * * * * * * * * * * * * *
             C
             C                 STATE A
             C
ISN 0019             DATA GDAY(16),CORECT(16) /45.,3.50                  /
ISN 0020             DATA GDAY(17),CORECT(17) /50.,3.25                  /
ISN 0021             DATA GDAY(18),CORECT(18) /55.,3.00                  /
ISN 0022             DATA GDAY(19),CORECT(19) /60.,2.75                  /
ISN 0023             DATA GDAY(20),CORECT(20) /65.,2.50                  /
ISN 0024             DATA GDAY(21),CORECT(21) /70.,2.25                  /
ISN 0025             DATA GDAY(22),CORECT(22) /75.,2.00                  /
ISN 0026             DATA GDAY(23),CORECT(23) /80.,1.75                  /
ISN 0027             DATA GDAY(24),CORECT(24) /85.,1.50                  /
ISN 0028             DATA GDAY(25),CORECT(25) /90.,1.25                  /
ISN 0029             DATA GDAY(26),CORECT(26) /95.,1.00                  /
ISN 0030             DATA GDAY(27),CORECT(27) /100.,0.75                 /
ISN 0031             DATA GDAY(28),CORECT(28) /105.,0.50                 /
ISN 0032             DATA GDAY(29),CORECT(29) /110.,0.25                 /
ISN 0033             DATA GDAY(30),CORECT(30) /115.,0.00                 /
ISN 0034             DATA GDAY(31),CORECT(31) /120.,-0.25                /
```

```
ISN 0035          DATA GDAY(32),CORECT(32) /125.,-0.50            /
ISN 0036          DATA GDAY(33),CORECT(33) /130.,-0.75            /
ISN 0037          DATA GDAY(34),CORECT(34) /135.,-1.00            /
            C
            C
            C
            C * * * * * * * * * * * * * * * * * * * * * *  * * * * * * * * * * * * * *
            C
            C                     STATE B
            C
ISN 0038          DATA GDAY(35),CORECT(35) /142.,-0.75            /
ISN 0039          DATA GDAY(36),CORECT(36) /149.,-0.50            /
ISN 0040          DATA GDAY(37),CORECT(37) /156.,-.25             /
ISN 0041          DATA GDAY(38),CORECT(38) /163.,0.00             /
ISN 0042          DATA GDAY(39),CORECT(39) /170.,0.25             /
ISN 0043          DATA GDAY(40),CORECT(40) /177.,0.50             /
ISN 0044          DATA GDAY(41),CORECT(41) /184.,0.75             /
ISN 0045          DATA GDAY(42),CORECT(42) /191.,1.00             /
ISN 0046          DATA GDAY(43),CORECT(43) /198.,1.25             /
ISN 0047          DATA GDAY(44),CORECT(44) /205.,1.50             /
ISN 0048          DATA GDAY(45),CORECT(45) /212.,1.75             /
            C
            C
            C
            C * * * * * * * * * * * * * * * * * * * * * *  * * * * * * * * * * * * * *
            C
            C                     STATE C
            C
ISN 0049          DATA GDAY(46),CORECT(46) /216.,1.50             /
ISN 0050          DATA GDAY(47),CORECT(47) /220.,1.25             /
ISN 0051          DATA GDAY(48),CORECT(48) /224.,1.00             /
ISN 0052          DATA GDAY(49),CORECT(49) /228.,0.75             /
ISN 0053          DATA GDAY(50),CORECT(50) /232.,0.50             /
ISN 0054          DATA GDAY(51),CORECT(51) /236.,0.25             /
ISN 0055          DATA GDAY(52),CORECT(52) /240.,0.00             /
ISN 0056          DATA GDAY(53),CORECT(53) /244.,-0.25            /
ISN 0057          DATA GDAY(54),CORECT(54) /248.,-0.50            /
ISN 0058          DATA GDAY(55),CORECT(55) /252.,-0.75            /
ISN 0059          DATA GDAY(56),CORECT(56) /256.,-1.00            /
ISN 0060          DATA GDAY(57),CORECT(57) /260.,-1.25            /
ISN 0061          DATA GDAY(58),CORECT(58) /264.,-1.50            /
ISN 0062          DATA GDAY(59),CORECT(59) /268.,-1.75            /
ISN 0063          DATA GDAY(60),CORECT(60) /272.,-2.00            /
ISN 0064          DATA GDAY(61),CORECT(61) /276.,-2.25            /
ISN 0065          DATA GDAY(62),CORECT(62) /280.,-2.50            /
ISN 0066          DATA GDAY(63),CORECT(63) /284.,-2.75            /
ISN 0067          DATA GDAY(64),CORECT(64)/ 288.,-3.00/
ISN 0068          DATA GDAY(65),CORECT(65)/ 292.,-3.25/
ISN 0069          DATA GDAY(66),CORECT(66)/ 296.,-3.50/
ISN 0070          DATA GDAY(67),CORECT(67)/ 300.,-3.75/
```

```
ISN 0071          DATA GDAY(68),CORECT(68)/ 304.,-4.00/
ISN 0072          DATA GDAY(69),CORECT(69)/ 308.,-4.25/
            C
            C
            C
            C * * * * * * * * * * * * * * * * * * * * * *  * * * * * * * * * * * * * *
            C
            C                 STATE D
            C
ISN 0073          DATA GDAY(70),CORECT(70)/ 311.,-4.00/
ISN 0074          DATA GDAY(71),CORECT(71)/ 314.,-3.75/
ISN 0075          DATA GDAY(72),CORECT(72)/ 317.,-3.50/
ISN 0076          DATA GDAY(73),CORECT(73)/ 320.,-3.25/
ISN 0077          DATA GDAY(74),CORECT(74)/ 323.,-3.00/
ISN 0078          DATA GDAY(75),CORECT(75)/ 326.,-2.75/
ISN 0079          DATA GDAY(76),CORECT(76)/ 329.,-2.50/
ISN 0080          DATA GDAY(77),CORECT(77)/ 332.,-2.25/
ISN 0081          DATA GDAY(78),CORECT(78)/ 335.,-2.00/
ISN 0082          DATA GDAY(79),CORECT(79)/ 338.,-1.75/
ISN 0083          DATA GDAY(80),CORECT(80)/ 341.,-1.50/
ISN 0084          DATA GDAY(81),CORECT(81)/ 344.,-1.25/
ISN 0085          DATA GDAY(82),CORECT(82)/ 347.,-1.00/
ISN 0086          DATA GDAY(83),CORECT(83)/ 350.,-0.75/
ISN 0087          DATA GDAY(84),CORECT(84)/ 353.,-0.50/
ISN 0088          DATA GDAY(85),CORECT(85)/ 356.,-0.25/
ISN 0089          DATA GDAY(86),CORECT(86)/ 359., 0.00/
ISN 0090          DATA GDAY(87),CORECT(87)/ 362., 0.25/
ISN 0091          DATA GDAY(88),CORECT(88)/ 365., 0.50/
            C
            C      CONTROL VALUE TO DEFINE THE LAST ELEMENT OF THE TABLES
            C
ISN 0092             DATA INDLST /88/
ISN 0093          END
```

```
LEVEL 2 FEB 67                                OS/360   FORTRAN H                                        DATE  68.067/17.16.05

          COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=50,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID

  ISN 0002              SUBROUTINE CURT (DDAY,VALUE)                                              0563
                C       SUBROUTINE TO PICK UP AN EPHEMERIS SUN SENSOR VALUE                       0564
                C       THE ROUTINE ASSUMES TABLES DEFINING A STEP FUN3TTON HAVE BEEN             0565
                C       COMPILED ANO STORED IN /SPETS/ COMMON                                     0566
                C       THE ROUTINE SCANS THE BODY ARRAY LOOKING FOR THE FIRST DAY LESS           0567
                C       THAN THE INPUT DAY.  IT THEN USES THE INDEX FOUND BY THE SCAN TO          0568
                C       REFERENCE THE BORECT TABLE                                                0569
  ISN 0003              REAL * 8 ODAY, VALUF                                                      0570
  ISN 0004              OATA OAYOLD, INDOLD /366.,24./                                            0571
                C         AN ATTEMPT TO SPEED UP THE LOOK UP WE ARE KEEPING TRACK OF THE          0572
                C      OLO VALUE ON THE LAST CYCLE AND STARTING THE TLU FROM THE                  0573
                C       PEVIOUS DAY VALUE.                                                        0574
  ISN 0005             COMMON /SPETS/ BOAY(100),BORECT(100),INDEX                                 0575
  ISN 0006             OAY    =  DDAY                                                             0576
  ISN 0007             IF(OAYOLO .GT. DAY) INDOLO = 1                                             0577
  ISN 0009             II   = INOOLD + 1                                                          0578
  ISN 0010             OO 1      I=II,INDEX,1                                                     0579
  ISN 0011             J = I-1                                                                    0580
  ISN 0012             IF(OAY-BDAY(I)) 2,I,1                                                      0581
  ISN 0013           1 CONTINUE                                                                   0582
  ISN 0014              J = INDEX                                                                 0583
  ISN 0015           2  CONTINUE                                                                  0584
  ISN 0016             DAYDLD = BDAY(J)                                                           0585
  ISN 0017             INOOLO = J                                                                 0586
  ISN 0018             VALUE = BORECT(J)                                                          0587
  ISN 0019             RETURN                                                                     0588
  ISN 0020             ENO                                                                        0589
```

```
LEVEL 2 FEB 67                        OS/360  FORTRAN H                        DATE  68.067/17.16.08

        COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=50,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID

  ISN 0002            BLOCK DATA                                                        0590
              C       DATA SUBROUTINE THE STEP FUNCTIONS FOR AN EPHEMERIS SUN SENSOR    0591
              C      VALUE                                                              0592
  ISN 0003           COMMON /SPETS/ BDAY(100),BDRECT(100),INDEX                         0593
  ISN 0004           DATA BDAY(1),BDRECT(1)/0.,-2.0/                                    0594
  ISN 0005           DATA BDAY(2),BDRECT(2)/10.,0.9/                                    0595
  ISN 0006           DATA BDAY(3),BDRECT(3)/20.,-1.1/                                   0596
  ISN 0007            DATA BDAY(4), BDRECT(4) /30.,-1.0 /                               0597
  ISN 0008            DATA BDAY(5), BDRECT(5) /40.,-0.8 /                               0598
  ISN 0009            DATA BDAY(6), BDRECT(6) /50.,-0.5 /                               0599
  ISN 0010            DATA BDAY(7), BDRECT(7) /60.,-0.3 /                               0600
  ISN 0011            DATA BDAY(8), BDRECT(8) /61., 0.0 /                               0601
  ISN 0012            DATA BDAY(9), BDRECT(9) /98., 0.0 /                               0602
  ISN 0013            DATA BDAY(10),BDRECT(10)/99., 0.3 /                               0603
  ISN 0014            DATA BDAY(11),BDRECT(11)/109., 0.4/                               0604
  ISN 0015            DATA BDAY(12),BDRECT(12)/119., 0.6/                               0605
  ISN 0016            DATA BDAY(13),BDRECT(13)/129., 1.0/                               0606
  ISN 0017            DATA BDAY(14),BDRECT(14)/139., 0.8/                               0607
  ISN 0018            DATA BDAY(15),BDRECT(15)/149.,-0.8/                               0608
  ISN 0019            DATA BDAY(16),BDRECT(16)/159., 0.8/                               0609
  ISN 0020            DATA BDAY(17),BDRECT(17)/169.,-1.0/                               0610
  ISN 0021            DATA BDAY(18),BDRECT(18)/179.,-0.8/                               0611
  ISN 0022            DATA BDAY(19),BDRECT(19)/189.,-0.6/                               0612
  ISN 0023            DATA BDAY(20),BDRECT(20)/199.,-0.4/                               0613
  ISN 0024            DATA BDAY(21),BDRECT(21)/200., 0.0/                               0614
  ISN 0025            DATA BDAY(22),BDRECT(22)/314., 0.0/                               0615
  ISN 0026            DATA BDAY(23),BDRECT(23)/315., 0.5/                               0616
  ISN 0027            DATA BDAY(24),BDRECT(24)/325., 0.7/                               0617
  ISN 0028            DATA BDAY(25),BDRECT(25)/335., 1.0/                               0618
  ISN 0029            DATA BDAY(26),BDRECT(26)/345., 1.1/                               0619
  ISN 0030            DATA BDAY(27),BDRECT(27)/355., 1.6/                               0620
  ISN 0031            DATA BDAY(28),BDRECT(28)/365.,-2.0/                               0621
  ISN 0032            DATA INDEX /28/                                                   0622
  ISN 0033            END                                                               0623
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=50,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID
ISN 0002              SUBROUTINE GETINP
               C
ISN 0003             IMPLICIT REAL * 8 (A-H,O-Z)
               C
               C     BEWARE
               C            SOME ENTRIES IN COMMON/PARAMS/ ARE DIFFERENT FROM THEIR
               C           COUNTERPARTS IN COMMON/PARAMS/ IN MAIN PROGRAM
               C     BEWARE
               C
               C
ISN 0004             COMMON/PARAMS/FITHDY,CONST  , XLIMXD, XDISP, XMASS , CUT   , A
                    1             E     ,XI    , OL    , XLAMD , TEND  , ALONGR, DATE,
                    2             FORCCT,FORLAM, A1    , A2    , W     , I1    , I2,
                    4             I3    ,I4    , I5    , I6    , I7    , NUM
               C
ISN 0005              COMMON/DOVE/XLIMO
ISN 0006             COMMON/MAN/C1,C2,C3,DMAX,CONN,NAV,N
ISN 0007              COMMON/ESL/NEWFR
ISN 0008             COMMON/SUPSTF/   AB1,   AB2, OFFSET,    I8
               C
ISN 0009              LOGICAL PUNCH,CUTDWN,AVERAG,THIRD,LINEAR,LINRAN,PERFCT,STEP,
                    1 STPRAN, PRINT, NEWFR, EPH, EPHRAN ,SAIL1 , SAIL2, SAIL3
                    2, FLIPW,FLIPAG,TACSAL,SAIL4
               C
ISN 0010              NAMELIST /SETUPS/ NUM  ,CONST ,XLIMD,XLIMXD,XDISP,XMASS,CUT  ,I4,
                    1A,E,XI,OL,XLAMD,TEND,ALONGR,DATE,I2,PUNCH,I5,CUTOWN,I7,AVERAG,I1,
                    2THIRD,I3,LINEAR,LINRAN,PERFCT,STEP,STPRAN,PRINT,ITHDAY,NEWFR,
                    3 C1,C2,C3,EPH,EPHRAN ,SAIL1 , SAIL2, A1,A2, SAIL3
                    4,I8,IRUN,FLIPW,FLIPAG,TACSAL,SAIL4,OFFSET,NAV
               CC
               C
ISN 0011              DIMENSION VFI1(8,2),VFI2(6,2),VFI3(16,10),VFI4(14,2),VFI7(11,2)
                    1 , VFI6(16,6)
                    2, VFI8(16,6)
ISN 0012              DIMENSION XVFI11(8),XVF(12(8),XVFI21I6) ,XVFI31(16),XVFI32I16),
                    1 XVFI33(16),XVF(34(16),XVFI35(16),XVFI36I16),XVFI37(16),XVFI38I16
                    2),XVFI39I16),XVFI41(14),XVFI42(14),XVFI71(11),XVFI72I11),XVFI22I6)
                    3 ,XVFI61I16),XVFI60(8) , XVI310(16)
                    4, XVFI81(16),XVFI82(16),XVFI80(16)       , XVFI62I16)
                    5,XVFI83(16)
               C
               C
ISN 0013              EQUIVALENCE (XVFI1I(1),VFII(I,1)), (XVFI12(1),VFI1(1,2)),
                    1 (XVFI21(1),VFI2(1,1)), (XVFI22(I),VFI2(1,2)),
                    2 (XVFI31(1),VF(3(1,1)),IXVFI32(1),VFI3(I,2)),
                    3  (XVF(33(I),VFI3(1,3)),(XVFI34(1),VFI3(I,4)),
                    4 (XVF(35(1),VF(3(1,5)), (XVFI36(I),VFI3(I,6)) ,
                    5  (XVFI71(1),VF(7(1,1)), (XVFI72(1), VFI7(1,2)),
                    6  (XVFI41(I), VFI4I1,I)) , (XVFI42(1), VFI4(1,2)),
```

```
                7(XVF(37(1), VFI3(1,7)), (XVFI38(1),VFI3(1,8)) ,
                8 (XVFI39(1),VFI3(1,9)), (XVI310(1),VFI3(1,10)),
                9 (XVFI61(1), VFI6(1,1)),(XVFI6D(1),VFI6(1,3))
                2,(XVFI81(1),VFI8(1,1)), (XVFI82(1),VFI8(1,2))
                1, (XVFI80(1),VFI8(1,4))        ,(XVFI62(1),VFI6(1,2))
                3,(XVFI83(1),VFI8(1,3))
          C
          C      R IS ENTERED IN KM,DR IN KM/SEC,THETA IN DEGREES,DTHETA IN
          C      DEGREES PER DAY,XLAM IN DEGREES,DXLAM IN DEGREES PER DAY
          C      TEND IS ENTERED IN SOLAR DAYS
          C       XLAM MEASURED EAST OF GREENWICH
          C      NAV  NUMBER OF DAYS TO AVERAGE XTEMP OVER FOR NEWER
          C
          C
          C         WHEN I1 = 1, Z6 CALC IS USED AND WHEN I1 = 2, Z7 CALC IS USED
          C         WHEN I2 = 1, PUNCH OUTPUT IS EXECUTED)))WHEN I2 = 2, NO PUNCHOU
          C         I3 CONTROLS SUN CORRECTION    I3 = 1  NO CORRECTION
          C                                       I3 = 2 CORRECTED LINEARLY
          C                                       I3 = 3 CORRECTED LINEARLY W/ RANDO
          C                                       I3 = 4 CORRECTED PERFECTLY
          C                                       I3 = 5 CORRECT BY STEP FUNCTION
          C                                       I3 = 6 CORRECTED BY S-F W/ RANDOM
          C                                       I3 =7 CORRECT BY EPHEMERIS SUN SEN
          C                                       I3 = 8 CORRECT BY EPHEMERIS SS W/R
          C                                       I3 = 9 1ST SUN SAIL CORRECTION
          C                                              CONSTANT FORCE ALONG LAMDA
          C                                       I3 =10 SOLAR SAIL3 FIXED SAIL
          C         IF I4 = 1 XJ22=-1.816E-6 AND XLAM22 = -15.0*.01745
          C         IF I4 = 2 XJ22=-1.700E-6 AND XLAM22 = -19.0*.01745
          C         IF I5 = 1, NO FORCE CUTDOWN
          C         IF I5 = 2, FORCE CUT BY INPUT NO. CUT
          C         I6   =  1  SOLAR SAIL2
          C      I6    =  2   SAIL IS TACKED
          C      IF I8 = 1,  SAIL IS FLIPPED WITH THE ORBIT
          C      IF I8 = 2, SAIL IS FLIPPED AGAINST THE ORBIT
          C      IF I8 = 3, SAIL IS FLIPPED BY LOGIC IN SATELLITE
          C
          C         CHARACTER TO BE PUNCHED TO SIGNAL THE END OF THE DATA OUTPUT
ISN 0014            DATA ZAP/-1./
          C
          C         VARIABLE FORMATS USED FOR THE OUTPUT
          C
ISN 0015         DATA XVFI83 /'(45H SAIL IS FLIPPED USING LOGIC IN THE SATELLITE)'/
ISN 0016         DATA XVFI82/'(33H SAIL IS FLIPPED AGAINST THE ORBIT)'/
ISN 0017         DATA XVFI81 /'(31H SAIL IS FLIPPED WITH THE ORBIT)'/
ISN 0018         DATA XVFI80/'(10H          )'/
ISN 0019          DATA XVFI72    /44H(39H PLOTTING XLAM AT END OF EVERY 10TH DAY)/
ISN 0020          DATA XVFI71    /43H(38H PLOTTING AVERAGE OF XLAM FOR 10TH DAY)/
ISN 0021         DATA XVFI62 /'(12H TACKED SAIL)'/
ISN 0022          DATA XVFI61 /25H(20H SOLAR SAIL SCHEME 2)/
```

```
ISN 0023          DATA XVFI60 /25H(20H                          )/
ISN 0024          DATA XVFI42   /56H(17H XJ22 = -1.700E-6 10X, 24H XLAM22 = -19.0 *
                1 .01745 )/
ISN 0025          DATA XVFI41   /56H(17H XJ22 = -1.816E-6 10X, 24H XLAM22 = -15.0 *
                I .01745 )/
ISN 0026         DATA XVI310 /30H(25H FIXED SAIL   SOLAR SAIL3)/
ISN 0027         DATA XVFI39 /44H(39H SOLAR SAIL1 CONSTANT FORCE ALONG LAMDA)/
ISN 0028         DATA XVFI38 /58H(53H SUN SENSOR CORRECTION, RANDOM ERROR OF +/-.5
                1DEGREES)/
ISN 0029         DATA XVFI37 /27H(22H SUN SENSOR CORRECTION)/
ISN 0030          DATA XVFI36   /61H(56H STEP FUNCTION CORRECTION, RANDOM ERROR OF
                1+/-.5 DEGREES)/
ISN 0031          DATA XVFI35   /30H(25H STEP FUNCTION CORRECTION)/
ISN 0032          DATA XVFI34   /25H(20H CORRECTED PERFECTLY)/
ISN 0033          DATA XVFI33   /58H(53H LINEAR SUN CORRECTION, RANDOM ERROR OF +/-
                1.5 DEGREES)/
ISN 0034          DATA XVFI32   /27H(22H LINEAR SUN CORRECTION)/
ISN 0035          DATA XVFI31   /23H(18H NO SUN CORRECTION)/
ISN 0036          DATA XVFI22   /21H(16H NO PUNCH OUTPUT)/
ISN 0037          DATA XVFI21   /18H(13H PUNCH OUTPUT)/
ISN 0038          DATA XVFI12   /30H(25H 3RD ORDER HARMONICS USED)/
ISN 0039          DATA XVFI11   /30H(25H 2ND ORDER HARMONICS USED)/
            C
            C
ISN 0040      17 FORMAT(8H XDISP =,F10.2,4X,8H XMASS =,F10.2,3X,9H ALONGR =,F10.2,
                1 'OFFSET =', F10.2)
ISN 0041      20 FORMAT( 4H A = ,1PE15.6, 4H E = ,1PE15.5, 4H I = ,1PE15.6,
                1 4H O = ,1PE15.6, 7H XLAM = , 1PE15.5,  7H TEND = ,1PE15.6)
ISN 0042      31 FORMAT(6H1NUM=  ,I5,9H CONST =  ,1PE15.6, 9H XLIMD =   ,1PE15.6,
                110H XLIMXD =   , 1PE15.6)
ISN 0043      39 FORMAT (17H NO FORCE CUTDOWN  )
ISN 0044      40 FORMAT (17H FORCE CUT BY     ,F10.4   )
ISN 0045      42 FORMAT (6H NUM =,I3,5X,8H CONST =,E10.2,5X,8H XLIMD =,E10.2)
ISN 0046      43 FORMAT (9H XLIMXD =,E10.4)
ISN 0047      44 FORMAT (4H A =,E10.2,4X,4H E =,F10.2,4X,5H XI =,F10.2,4X,5H DL =,
                1E10.2)
ISN 0048      45 FORMAT (7H XLAM =,F10.2,4X,7H TEND =,E10.2)
ISN 0049      50 FORMAT (' RUN NUMBER= ', I4)
ISN 0050      51 FORMAT ('1')
ISN 0051      52 FORMAT ('OYR=',I5,5X,'DAY=',I5)
ISN 0052     101 FORMAT (7H DATE =,F10.2,5X,3HI1=,I3,3X,3HI2=,I3,3X,3HI3=,I3,3X,
                1                        3HI4=,I3,3X,3HI5=,I3,
                2        /22X,           3HI6=,I3,3X,3HI7=,I3,3X,3HI8=,I3)
ISN 0053     888  FORMAT(' NEW FORCE ROUTINE USED')
ISN 0054     999  FORMAT(' ORIGINAL FORCE ROUTINE USED')
            C
            C
            C     THESE CONSTANTS ARE NOT INITIALIZED BY DATA STATEMENTS BECAUSE
            C             WE COULD LOOP BACK TO THIS POINT IN THE PROGRAM AND WE WA
            C             WANT THEM RESET AT THAT POINT.
```

```
               C              ANO MOST OF THEM APPEAR IN A LABELED COMMON....
               C
               C
               C
ISN 0055             A      = 42165.5DO
ISN 0056             A1     = .9DO
ISN 0057             A2     = .1DO
ISN 0058             ALONGR= 0.DO
ISN 0059             CONST =-4.45D-8
ISN 0060             CUT    = 0.DO
ISN 0061             DATE   = 0.DO
ISN 0062             E      = 0.DO
ISN 0063             FORCCT    = 0.0DO
ISN 0064             FORLAM    = 1.0DO
ISN 0065             IRUN   =  0
ISN 0066             ITHDAY    = 10
ISN 0067             NAV=4
ISN 0068             NUM    =  6
ISN 0069             OFFSET=  -90.0DO
ISN 0070             OL     = 0.DO
ISN 0071             TEND   = 1800.DO
ISN 0072             XDISP = 70.DO
ISN 0073             XI     = 0.DO
ISN 0074             XLAMO = 300.DO
ISN 0075             XLIMD = 297.DO
ISN 0076             XLIMXD= 350.DO
ISN 0077             XMASS = 132.DO
               C
               C
               C     INITIAL CONDITIONS
               C
               C     PRINT =  .TRUE.          CONTROL NAMELIST PRINT
               C     I1     =  I               SECOND ORDER HARMONICS
               C     I2     =  1               PUNCH = .TRUE.
               C     I3     =  1               NO CORRECTION
               C     I4     =  2
               C     I5     =  1               NO CUT DOWN
               C     I6     =  3
               C     I7     =  2               NOT AVERAGED
               C     I8     =  4   SAIL IS NOT FLIPPED
               C
               C
ISN 0078              I1       =    1
ISN 0079              I2 = 1
ISN 0080              I3       =    1
ISN 0081              I4 =2
ISN 0082              I5       =    1
ISN 0083             I6     = 3
ISN 0084              I7       =    2
ISN 0085             I8     =  4
```

```
              C                                                                                   0821
ISN 0086             AVERAG  =    .TRUE.                                                          0822
ISN 0087             CUTDWN  =    .FALSE.                                                         0823
ISN 0088             EPH     =    .FALSE.                                                         0824
ISN 0089             EPHRAN  =    .FALSE.                                                         0825
ISN 0090            FLIPAG        =          .FALSE.                                              0826
ISN 0091            FLIPW    =    .FALSE.                                                         0827
ISN 0092             LINEAR  =    .FALSE.                                                         0828
ISN 0093             LINRAN  =    .FALSE.                                                         0829
ISN 0094             NEWFR=.FALSE.                                                                0830
ISN 0095             PERFCT  =    .FALSE.                                                         0831
ISN 0096             PRINT = .TRUE.                                                               0832
ISN 0097             PUNCH = .TRUE.                                                               0833
ISN 0098             SAIL1 =      .FALSE.                                                         0834
ISN 0099            N =1
ISN 0100             SAIL2   =    .FALSE.                                                         0835
ISN 0101             SAIL3   =   .FALSE.                                                          0836
ISN 0102            SAIL4 =  .FALSE.                                                              0837
ISN 0103             STEP    =    .FALSE.                                                         0838
ISN 0104             STPRAN  =    .FALSE.                                                         0839
ISN 0105            TACSAL = .FALSE.                                                              0840
ISN 0106             THIRD   =    .FALSE.                                                         0841
              C                                                                                   0842
              C      READ THE NEW CONTROL VALUES                                                  0843
              C                                                                                   0844
ISN 0107             READ (5,SETUPS)                                                              0845
ISN 0108            C1     =  CONST                                                               0847
ISN 0109            C2     =  CONST / 4.D0                                                        0848
ISN 0110            C3     =  CONST                                                               0849
ISN 0111            CONN=CONST/32
ISN 0112             DMAX=DABS(C1)/25.D0                                                          0846
ISN 0113            XLAM   =  XLAMD                                                               0850
              C                                                                                   0851
              C      CHECK NUM FOR THE END OF DATA BEFORE WE GO THROUGH THE OUTPUT                0852
              C              PHASE OF THIS SECTION                                                0853
ISN 0114               IF(NUM .LT.  0)      GO TO 10                                              0854
              C                                                                                   0855
              C      MODIFY THE CONTROL VARIABLES AS A FUNCTION OF THE INPUT                      0856
              C                                                                                   0857
              C                                                                                   0858
ISN 0116             IF(THIRD)    I1=2                                                            0859
              C                                                                                   0860
              C                                                                                   0861
              C      DEFINE THE PUNCH VARIABLE FOR THE OUTPUT SUBROUTINE                          0862
              C              ''FITOUT''                                                           0863
ISN 0118             IF(PUNCH)    I2  =1                                                          0864
ISN 0120             IF(I2 .EQ. 1)      PUNCH = .TRUE.                                            0865
ISN 0122             CALL SETOUT(PUNCH)                                                           0866
              C                                                                                   0867
ISN 0123             IF(AVERAG)   I7=1                                                            0868
```

```
ISN 0125          IF(CUTDWN)  I5=2
ISN 0127          IF(EPH) I3=7
ISN 0129          IF(EPHRAN) I3=8
ISN 0131         IF(FLIPAG)   I8  =  2
ISN 0133         IF(FLIPW)  I8  =  1
ISN 0135          IF(LINEAR)  I3=2
ISN 0137          IF(LINRAN)  I3=3
ISN 0139          IF(PERFCT)I3=4
ISN 0141          IF (SAIL1) I3 = 9
ISN 0143          IF (SAIL2)  I6 = 1
ISN 0145         IF  (SAIL3)  (3 = 10
ISN 0147         IF (SAIL4) I8 = 3
ISN 0149         IF (SAIL4) I6 = 1
ISN 0151          IF(STEP)    I3=5
ISN 0153          IF(STPRAN)  I3=6
ISN 0155         IF(TACSAL) I6  =  2
            C
            C     PRINT OUT THE NAMELIST AS A FUNCTION OF THE PARAMETER PRINT
            C
ISN 0157         WRITE (6,51)
ISN 0158          IF(PRINT) WRITE(6,SETUPS)
ISN 0160         CALL TIMOAT (IYR,IOAY)
ISN 0161         WRITE (6,52) IYR,IOAY
ISN 0162          FITHDY  =   ITHOAY
            C
            C
ISN 0163         WRITE (6     ,31)NUM,CONST,XLIMD,XLIMXD
ISN 0164         WRITE (6, 17) XDISP, XMASS, ALONGR,OFFSET
ISN 0165         WRITE (6     ,20)A,E,XI,OL,XLAM,TEND
ISN 0166         WRITE (6,101) DATE, I1, I2, I3, I4, I5, I6, I7,I8
            C
ISN 0167          IF(.NOT. PUNCH) GO TO 9
            C
ISN 0169       18 WRITE (7, 42) NUM, CONST, XLIMD
ISN 0170          WRITE(7, 43) XLIMXD
ISN 0171          WRITE(7, 17) XOISP, XMASS, ALONGR , OFFSET
ISN 0172          WRITE (7, 44) A, E, XI, OL
ISN 0173          WRITE (7, 45) XLAM, TEND
ISN 0174         WRITE (7,101) OATE, I1, I2, I3, I4, I5, I6, I7,I8
            C
ISN 0175          IF(NEWFR)GO TO 12
ISN 0177          WRITE(7,999)
            C
ISN 0178          GO TO 9
ISN 0179     12   WRITE(7,888)
ISN 0180       9  CONTINUE
ISN 0181          CALL R(TOUT(VFI1(1,I1))
ISN 0182          CALL R(TOUT(VFI2(1,I2))
ISN 0183          CALL RITOUT(VFI3(1,I3))
ISN 0184          CALL RITOUT(VF(4(1,I4))
```

```
ISN 0185         IF (SAIL4) I6 = 0
ISN 0187          CALL RITOUT (VFI6 (1,I6))
           C
ISN 0188          IF(I5-1)1,1,3
           C
ISN 0189        1 WRITE(6,39)
ISN 0190          IF(PUNCH) WRITE(7,39)
ISN 0192          GO TO 4
           C
ISN 0193        3 WRITE(6,40)CUT
ISN 0194          IF(PUNCH) WRITE(7,40)CUT
ISN 0196        4 CONTINUE
ISN 0197          CALL RITOUT(VFI7(1,I7))
ISN 0198         CALL RITOUT (VFI8(1,I8))
ISN 0199         WRITE (6,50) IRUN
ISN 0200         IF (PUNCH) WRITE(7,50) IRUN
ISN 0202         IF (SAIL4) I6 = 1
ISN 0204         IF(I6.EQ.3) I6=0
ISN 0206         IF(I8.EQ.4) I8=0
ISN 0208          RETURN
           C
ISN 0209      10 IF (PUNCH)  WRITE(7,11) ZAP,ZAP
ISN 0211      11 FORMAT(2F20.7)
           C
ISN 0212         RETURN
ISN 0213          END
```

```
LEVEL 2 FEB 67                          OS/360  FORTRAN H                            DATE  68.067/17.16.23

         COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=50,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID

  ISN 0002            SUBROUTINE MINMAX (X, N, XMIN, XMAX )                              0943
  ISN 0003            OIMENSION X(1500)                                                  0944
  ISN 0004            DOUBLE PRECISION X, XMIN, XMAX                                     0945
  ISN 0005            XMIN = X(1)                                                        0946
  ISN C006             XMAX = X(1)                                                       0947
  ISN 0007            DO 10 I=2,N                                                        0948
  ISN 0008            IF (X(I)-XMIN) 1,2,2                                               0949
  ISN 0009          1  XMIN = X(I)                                                       0950
  ISN 0010      2     IF (XMAX-X(I)) 3,10,10                                             0951
  ISN 0011          3 XMAX = X(I)                                                        0952
  ISN 0012         10 CONTINUE                                                           0953
  ISN 0013            RETURN                                                             0954
  ISN 0014            END                                                                0955
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=50,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID

ISN 0002             SUBROUTINE NEWFOR(XLIMO,XTEMP,CON)                               0956
           C IN THIS PROGRAM WE WILL CALCULATE A FORCE,CON.  THIS FORCE WILL DEPEND   0957
           C ON WHICH REGION THE VARIABLE XTEMP IS IN.  WHEN WE GET THE SEQUENCE      0958
           C OF REGIONS +A-A+A  OR  -A+A-A WE WILL CALCULATE THE FORCE DIFFERENTLY.   0959
           C                                                                          0960
ISN 0003            IMPLICIT REAL*8 (A-H,O-Z)
ISN 0004            DIMENSION COUNT(1800)   , XT(35)
ISN 0005            COMMON/MAN/C1,C2,C3,OMAX,CONN,NAV,N
ISN 0006             COMMON/EL/CAVE,J1,J2,J3,JANE,K,LAO                               0963
ISN 0007            DATA NI /0/
           C                                                                          0965
           C  THIS SUBROUTINE WORKS WITH QUANTITIES IN DEGREES(UNITS)                  0966
ISN 0008             XTEMP = XTEMP / .17453292519943290D-01                           0967
ISN 0009            NI=NI+1
ISN 0010            XT(NI)=XTEMP
ISN 0011            IF(N.EQ.1) GO TO 8
ISN 0013            CUN=CONN
ISN 0014            IF(NI.LT.NAV)RETURN
ISN 0016            SUM=0.0
ISN 0017            DO 3 J=1,NAV
ISN 0018          3 SUM=XT(J) + SUM
ISN 0019            XTEMP=SUM/NAV
ISN 0020            NI=0
ISN 0021          8 N=2
           C THIS SECTION WILL DETERMINE WHERE XTEMP LIES                             0968
ISN 0022             IF(XTEMP .GE.XLIMO) GO TO 10                                     0969
ISN 0024             GO TO 20                                                         0970
           C                                                                          0971
           C AT THIS POINT XTEMP IS GREATER THAN OR EQUAL TO XLIM                     0972
ISN 0025          10  R=-1.D0                                                         0973
ISN 0026             I=-1                                                             0974
ISN 0027            IF(XTEMP.GE.(XLIMO+4.00))GO TO 30
ISN 0029             IF(XTEMP.GE.(XLIMO+2.00))GO TO 50                                0975
ISN 0031             GO TO 70                                                         0977
           C                                                                          0978
           C  AT THIS POINT XTEMP IS LESS THAN XLIMO                                  0979
ISN 0032          20  R=1.0D0                                                         0980
ISN 0033             I=1                                                              0981
ISN 0034            IF(XTEMP.LE.(XLIMO-4.00))GO TO 30
ISN 0036             IF(XTEMP.LE.(XLIMO-2.00))GO TO 50                                0983
ISN 0038             GO TO 70                                                         0984
           C                                                                          0985
           C  XTEMP IS IN REGION C.  SET J1=3 OR -3, FORCE=C3 OR -C3 AND J2,J3,=0.    0986
           C  TEST TO SEE IF WE HAVE BEEN IN C AWHILE                                 0987
ISN 0039         30   INEW = I*3                                                      0988
ISN 0040             IF(INEW .EQ. JI)GO TO 600                                        0989
ISN 0042             J1 = INEW                                                        0990
ISN 0043             CONN = R*C3                                                      0991
ISN 0044             K = I                                                            0992
```

```
ISN 0045          COUNT(K) = CONN
ISN 0046          J2 = 0
ISN 0047          J3 = 0
ISN 0048          JANE = 0
ISN 0049          CAVE = 0.D0
ISN 0050          LAO = 0
ISN 0051          GO TO 600
            C
            C  WE ARE IN REGION B.  HAVE WE CHANGED REGIONS--IF SO,UPDATE AND SET
            C  FORCE = C2 OR -C2.
ISN 0052      50   INEW = I*2
ISN 0053          IF(INEW .EQ. J1) GO TO 600
ISN 0055          IF(J1 .EQ.0) K=0
ISN 0057          CONN = R*C2
ISN 0058          GO TO 205
            C
            C  WE ARE IN REGION A, ANO MUST OECIOE WHAT VALUE TO GIVE THE FORCE.
            C  CHECK TO SEE IF WE ARE IN THE SAME REGION. HAVE WE CHANGED REGIONS?
ISN 0059      70   INEW = I*1
ISN 0060          IF(J1 .EQ. 0)K=0
ISN 0062          IF(INEW  .EQ. J1) GO TO 600
            C  DIO WE JUST GO FROM B TO A.
ISN 0064          IF(IABS(J1) .EQ. 2)GO TO 200
            C  TEST FOR INITIAL SEQUENCE
ISN 0066          IF(JANE .EQ. 0 )GO TO 180
            C
            C  HAVE PASSED THE INITIAL SEQUENCE ANO CHANGED REGIONS. TEST FOR MAX DA
            C  ANO ZERO BUMP SECTION.
ISN 0068     170   IF(IABS(INEW*J1*J2) .NE. 1 .ANO. LAD .EQ. 0) GO TO 190
ISN 0070          IF(IABS(INEW*J1*J2) .NE. 1 .AND. LAD .EQ. 5) GO TO 181
ISN 0072          IF(OABS(CONN) .NE. 0.00 .AND. OABS(CONN) .LT. OMAX) GO TO 200
ISN 0074          IF(LAD .EQ. 5) GO TO 5
            C
            C  IN A AND SEQUENCING. JUST CHANGED REGIONS. MAX DAMP AND ZERO BUMP TAK
            C  TAKEN CARE OF. ALSO COMING FROM B. (INEW*J1*J2=1)
ISN 0076     110   IF(DABS(COUNT(K)) .NE. OABS(COUNT(K-1)))GO TO 135
            C
            CIF WE GET HERE IT MEANS WE ARE READY TO DIVIDE BY 2
ISN 0078     130  CONN = -CONN/2.D0
ISN 0079          CAVE = CONN
ISN 0080          IF(DABS(CONN) .LT. OMAX)LAD=5
ISN 0082          GO TO 205
            C
            C  ABSOLUTE VALUE REMAINS THE SAME IF J3 .NE. 2.
ISN 0083     135   IF(IABS(J3) .EQ. 2) GO TO 130
ISN 0085          CONN = -CONN
ISN 0086          CAVE = CONN
ISN 0087          GO TO 205
            C
            C  TEST HERE FOR INITIAL SEQUENCE.
```

```
ISN 0088      130  IF(IABS(INEW*J1*J2) .EQ. 1 )GO TO 207
            C  WE ARE IN A BUT HAVE NOT REACHED INITIAL SEQUENCE.
ISN 0090           CONN = R*C1
ISN 0091           GO TO 205
            C
            C  IN MAX DAMP ZERO BUMP SECTION AND INEW*J1*J2 NE 1
ISN 0092      181  IF(CONN .EQ. 0.D0) GO TO 5
ISN 0094           GO TO 200
            C
            C  INITIAL SEQUENCE JUST REACHED.  MEANS CHANGE OF REGION .
ISN 0095      207  JANE = 1
ISN 0096           CONN = R*C1/2.D0
ISN 0097           CAVE = CONN
ISN 0098           GO TO 205
            C
            C  IN A AFTER THE INITIAL SEQUENCE. INEW*J1*J2 .NE. 1 .
ISN 0099      190 CONN=-CAVE
ISN 0100           GO TO 205
            C
            C  IN ZERO BUMP AND MAXIMUM DAMP SECTION
ISN 0101        5  CONN = R*C1/32.D0
ISN 0102           GO TO 205
            C
            C  FROM B TO A
ISN 0103      200  CONN = 0.0D0
            C  REGIONS HAVE JUST CHANGED.  UPDATE SEQUENCE
ISN 0104      205  K = K+1
ISN 0105           J3 = J2
ISN 0106           J2 = J1
ISN 0107           J1 = INEW
ISN 0108           COUNT(K) = CONN
            C
ISN 0109      600  CON = CONN
ISN 0110           RETURN
ISN 0111           END
```

```
          COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=50,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID
   ISN 0002            SUBROUTINE  SETOUT(PUNCH)                                              1078
                C      SUBROUTINE TO OUTPUT ON LUN 6 AND THE PUNCH TAPE  LUN 7                1079
                C              THE VARIABLE FORMAT INFORMATION                                1080
   ISN 0003            LOGICAL PUNCH,CARDS                                                    1081
                C                                                                             1082
                C      ENTRY POINT TO DEFINE THE LOGICAL VARIABLE CARDS FOR THE SECOND        1083
                C              ENTRY POINT   RITOUT                                           1084
   ISN 0004            CARDS   =   PUNCH                                                      1085
   ISN 0005            RETURN                                                                 1086
                C                                                                             1087
                C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *       1088
                C                                                                             1089
                C      ENTRY POINT TO DO THE ACTUAL WRITE AND PUNCH                           1090
                C                                                                             1091
   ISN 0006            ENTRY RITOUT (V)                                                       1092
   ISN 0007            DIMENSION V(18)                                                        1093
   ISN 0008            WRITE(6,V)
   ISN 0009            IF(CARDS)   WRITE(7,V)                                                 1095
   ISN 0011            RETURN                                                                 1096
   ISN 0012            END                                                                    1097
```

```
LEVEL 2 FEB 67                           DS/360  FORTRAN H                              DATE  58.067/17.16.35

      -COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=50,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID

 ISN 0002             SUBROUTINE SSL                                                  1098
 ISN 0003             DOUBLE PRECISION X                                              1099
 ISN 0004             DOUBLE PRECISION CIND                                           1100
 ISN 0005             X=CIND(0,X,0.000)                                               1101
 ISN 0006             RETURN                                                          1102
 ISN 0007             END                                                             1103
```

```
      COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=50,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID
ISN 0002            SUBROUTINE TACK(XLAM)
              C
              C     WRITTEN BY ELLEN SWENSON X5309
ISN 0003            IMPLICIT REAL * 8 (A-H,O-Z)
ISN 0004            DOUBLE PRECISION LOBSTR
ISN 0005            LOGICAL CHECK
ISN 0006            CHECK =       .FALSE.
              C
ISN 0007            COMMON/Z8/XLIM   , CONST  , XLIMX , CUT    , XNUM   , FUEL   ,
                   1         XX     , WS     , COSI   , ESUN   , SINI   , TEL    , JFK    ,
                   2         LLL    , M      , K      , NUM    , I3     , I5
ISN 0008            COMMON/PARAMS/FITHDY,CONSTI, XLIMXD, XDISP, XMASS , CUTBY , A     ,
                   1              E      ,XI     , DL     , XLAMD , TENO  , ALONGR, DATE,
                   2                 FORCCT,FORLAM, A1     , A2     , W      , I1     , I2,
                   3                 I3T    ,I4     , I5T    , I6     , I7     , NUMIN
              C
ISN 0009            COMMON/SUPSTF/   AB1,    AB2, OFFSET,     I8
              C     OBJECT
              C           TO OBTAIN MAXIMUM FORCEF ALONG ORBITAL TANGENT
              C
              C     UNDERLYING THEORY
              C           LARRY BLACK'S IDEA-- OPTIMUM POSITION FOR SAIL IS ALONG
              C           BISECTOR OF THE ANGLE BETWEEN THE PROJECTION OF THE SUN IN T
              C           ORBITAL PLANE AND THE ORBITAL TANGENT
              C     THERE IS PROVISION FOR CHANGING THE POSITION OF SAIL
              C           ROTATE SAIL 'OFESET'DEGREES FROM TACKED POSITION
              C     THERE IS AN ENTRY CHAN AND I8 VARIABLES
              C           CHAN PLACES SAIL ON THE RADIUS
              C     I8 FLIPS THE SAIL THREE WAYS
              C           FLIPS SAIL WITH ORBIT
              C            FLIPS SAIL AGINST ORBIT
              C           FLIPS SAIL EVERYWHERE
              C
              C BOTH SIDES OF THE SAIL ARE REFLECTIVE
ISN 0010            AB1    =       .1D0
ISN 0011            AB2    =   .1D0
              C
              C      ALONGR IS FORCE ALONG R
              C      FORCCT IS FORCE ALONG CTHETA
              C      FORLAM IS FORCE ALONG LAMDA
              C
              C     DARCOS(DOTPRO)IS ANGLE BETWEEN NORMAL TO THE SAIL AND THE SUN
              C     CTHETA IS ANGLE BETWEEN ARIES AND RADIUS
              C
ISN 0012       300  CONTINUE
              C
              C  GEOCENTRIC CORRDINATES OF NORMAL TO RADIUS
ISN 0013         1   CTHETA = XLAM + W * TEL
ISN 0014            XNORMA    =    DSIN(CTHETA)
```

```
ISN 0015          YNORMA    =    -OCOS(CTHETA)
ISN 0016          ZNORMA    =    0.000
               C
               C   GEOCENTRIC COORDINATES OF SUN
               C       V = TRUE ANOMALY (MOULYON P171)     XM = MEAN ANOMALY
ISN 0017               DAY = TEL/86400.DO
ISN 0018           2   XM =   (.98560000 * DAY -3.392900) * XX
ISN 0019               V    = XM + 2.00 * ESUN *OSIN(XM) + 5.00 * ESUN *ESUN * DSIN(2.00
                  1        * XM) / 4.DO
               C
ISN 0020               U    = V  +  WS
ISN 0021               SINU= DSIN (U)
               C
ISN 0022               XSUN= OCOS (U)
ISN 0023               YSUN= SINU * COSI
ISN 0024               ZSUN= SINU * SINI
               C
               C GEOCENTRIC COORDINATES OF THE  ORBITAL TANGENT
ISN 0025          XFORCE    =    -XNORMA
ISN 0026          YFORCE    =    -YNORMA
ISN 0027          ZFORCE    =    ZNORMA
               C
               C GEOCENTRIC COORDINATES OF PROJECTION OF THE SUN IN THE ORBITAL PLNE
ISN 0028          XMAINE =OSQRT( XSUN**2 + YSUN**2      )
ISN 0029          XPROJ =       XSUN/XMAINE
ISN 0030          YPROJ =       YSUN/XMAINE
               C
ISN 0031          IF (CHECK) GO TO 310
               C
               C COORDINATES OF SAIL ITSELF
ISN 0033          XTEMP =       XFORCE + XPROJ
ISN 0034          YTEMP =       YFORCE + YPROJ
               C  COORDINATES OF NORMALIZED SAIL VECTOR
               C
ISN 0035          YMAINE=   DSQRT ( XTEMP**2 + YTEMP**2)
ISN 0036          XTEMP =  XTEMP / YMAINE
ISN 0037          YTEMP =  YTEMP / YMAINE
               C
               C COORDINATES OF SAIL ITSELF
ISN 0038          SINOFF=       OSIN(OFFSET)
ISN 0039          COSOFF=       OCOS(OFFSET)
ISN 0040          XRESAL=       XTEMP* COSOFF + YTEMP*SINOFF
ISN 0041          YRESAL=       YTEMP * COSOFF - XTEMP * SINOFF
ISN 0042          XSAIL =       YRESAL
ISN 0043          YSAIL = -XRESAL
ISN 0044          GO TO 3
               C
ISN 0045        310  CONTINUE
ISN 0046          XSAIL = XNORMA
ISN 0047          YSAIL = YNORMA
```

```
ISN 0048          XRESAL =      -YSAIL
ISN 0049          YRESAL =      XSAIL
             C
ISN 0050           3  DOTPRO = XSAIL *XSUN + YSAIL * YSUN
             C
             C ANGLE BETWEN SAIL AND RADIUS =  ANCHOR
ISN 0051          LOBSTR=       XFORCE*(-XSAIL) + YFORCE*(-YSAIL)
ISN 0052          ANCHOR=       DARCOS(LOBSTR)
ISN 0053          BOAT  =       DSIN(ANCHOR)
             C
ISN 0054          ROCK  =   DABS(LOBSTR)
ISN 0055          TRAP  =   DABS(DOTPRO)
ISN 0056          IF(DOTPRO .LE. 0.D0) AB1 = AB2
ISN 0058          POT   =   (2.000 - AB1) * DOTPRO**2
             C
             C ALOSAL IS FORCE ALONG SAIL DUE TO NON-PERFECT SAIL
ISN 0059          ALOSAL=     - AB1 * TRAP *(XSUN*XRESAL + YSUN*YRESAL)
             C
             C TO DETERMINE DIRECTION OF FORCE COMPONENTS OF ALONGR ANDFORLAM
ISN 0060          IF (IB .EQ.1 .AND. DOTPRO .LE. 0.D0) GO TO 40
ISN 0062          IF (IB .EQ. 2 .AND. DOTPRO .GE. 0.D0) GO TO 40
ISN 0064          IF (IB .EQ.3) GO TO 40
             C
             C          COD   FISH   HERRING   SALT
             C          ARE INITIALIZED TO +1  ABD MADE NEGATIVE WHEN NECESSARY
             C
             C     COD IS NEGATIVE WHEN
             C          DIRECTION OF SOLAR FORCE IS MORE THAN90 DEG FROM RADIUS
             C
             C     FISH IS NEGATIVE WHEN
             C          SAIL IS MORE THAN 90 DEGREES FROM RADIUS
             C
             C     HERING IS NEGATIVE WHEN
             C          DIRECTION OF SOLAR FORCE IS MORE THAN 90DEG FROM ORBITAL TAN
             C
             C     SALT IS NEGATIVE WHEN
             C          SAIL IS MORE THAN 90 DEGREES FROM ORBITAL TANGENT
             C
ISN 0066          COD   =       1.000
ISN 0067          FISH  =       1.000
ISN 0068          HERING=       1.000
ISN 0069          SALT  =       1.000
             C
             C     COORDINATES OF RADIUS
             C          X COORDINATE  -YNORMA
             C          Y COORDINATE   XNORMA
             C
             C COORDINATES OF SOLAR RADIATION FORCE
             C          TESTING ON Z-COMPONENT OF CROSS PRODUCT OF SAIL CROSS SUN
             C
```

```
ISN 0070          IF((XRESAL * YPROJ - XPROJ * YRESAL) .LE.0.00) GO TO 7
ISN 0072          XSORAF=      XSAIL
ISN 0073          YSORAF=      YSAIL
ISN 0074          GO TO 6
            C
ISN 0075        7 XSORAF=      -XSAIL
ISN 0076          YSORAF=      -YSAIL
            C
ISN 0077        6 CONTINUE
ISN 0078          IF((XSORAF*XFORCE + YSORAF*YFORCE).LE.0.0D0) HERING=-1.0D0
ISN 0080          IF((XRESAL * XFORCE +YRESAL * YFORCE) .LE.0.D0) SALT=-1.D0
ISN 0082          IF((XSORAF * (-YNORMA) + YSORAF * XNORMA).LE.0.0D0) COD=-1.0D0
ISN 0084          IF((-YNORMA)*XRESAL + XNORMA * YRESAL .LE. 0.D0) FISH=-1.D0
            C
            C
ISN 0086          FORLAM=      HERING*POT*ROCK + SALT*ALOSAL*BOAT
ISN 0087          ALONGR=      COD*POT*BOAT + FISH*ALOSAL*ROCK
ISN 0088          FORCCT=  -AB1 * TRAP* ZSUN
            C
ISN 0089          RETURN
            C
ISN 0090       40 CONTINUE
ISN 0091          DOTPR2=      DABS(ZSUN)
ISN 0092          DOTPR1=      XSUN * (-YSAIL) + YSUN * XSAIL
ISN 0093          FORLAM   =   AB1 * DOTPR2 * DOTPRO
ISN 0094          ALONGR   =   -AB1 * DOTPR2 * DOTPR1
            C
            C TO OBTAIN FORCCT TEST ON SIGN OF ZSUN
ISN 0095          IF(ZSUN .GE.0.D0) FORCCT = -(2.D0 - AB1)*DOTPR2**2
ISN 0097          IF(ZSUN .LE.0.D0) FORCCT =  (2.D0 - AB1)*DOTPR2**2
ISN 0099          RETURN
            C
ISN 0100          ENTRY CHAN(XLAM)
ISN 0101          ROTAAT=    XLAM + W*TEL
ISN 0102          CHECK =       .TRUE.
ISN 0103          GO TO 300
ISN 0104          END
```

```
LEVEL 2 FEB 67                                OS/360   FORTRAN H                                   DATE   68.067/17.16.46

          COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=50,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID

   ISN 0002              DOUBLE PRECISION FUNCTION CIND(K,A,B)                                  1290
   ISN 0003              DOUBLE PRECISION Y0(50),DY0(50),DY1(50),DY2(50),A,B,X,H                1291
   ISN 0004              DOUBLE PRECISION Y,DY,A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12           1292
   ISN 0005              DOUBLE PRECISION B1,B2,B3,B4                                           1293
   ISN 0006              IF (K) 1000,10,1000                                                    1294
   ISN 0007      10      IF (B) 30,20,30                                                        1295
   ISN 0008      20      J=1                                                                    1296
   ISN 0009              A1=.3333333333333333                                                   1297
   ISN 0010              A2=.1666666666666667                                                   1298
   ISN 0011              A3=.4166666666666667                                                   1299
   ISN 0012              A4=.6666666666666667                                                   1300
   ISN 0013              A5=.83333333333333330-1                                                1301
   ISN 0014              A6=1.916666666666667                                                   1302
   ISN 0015              A7=1.333333333333333                                                   1303
   ISN 0016              A8=.4166666666666666                                                   1304
   ISN 0017              A9=.375D0                                                              1305
   ISN 0018              A10=.7916666666666667                                                  1306
   ISN 0019              A11=.2083333333333333                                                  1307
   ISN 0020              A12=.41666666666666660-1                                               1308
   ISN 0021              B1=2.291666666666667                                                   1309
   ISN 0022              B2=2.458333333333333                                                   1310
   ISN 0023              B3=1.541666666666667                                                   1311
   ISN 0024              B4=.375D0                                                              1312
   ISN 0025              RETURN                                                                 1313
   ISN 0026      30      X=A                                                                    1314
   ISN 0027              H=B                                                                    1315
   ISN 0028              GO TO (100,150,150,100,150,100,150,100,300),J                          1316
   ISN 0029      100     CIND=X+H                                                               1317
   ISN 0030              GO TO 200                                                              1318
   ISN 0031      150     CIND=X                                                                 1319
   ISN 0032      200     J=J+1                                                                  1320
   ISN 0033      250     I=1                                                                    1321
   ISN 0034              RETURN                                                                 1322
   ISN 0035      300     CIND=X+H                                                               1323
   ISN 0036              GO TO 250                                                              1324
   ISN 0037      1000    Y=A                                                                    1325
   ISN 0038              DY=B                                                                   1326
   ISN 0039              GO TO (5000,1200,1300,1400,1500,1600,1700,1800,1900),J                 1327
   ISN 0040      1200    CIND=Y+H*DY                                                            1328
   ISN 0041              Y0(I)=Y                                                                1329
   ISN 0042              DY0(I)=DY                                                              1330
   ISN 0043              GO TO 3000                                                             1331
   ISN 0044      1300    CIND=Y0(I)+H*.5D0*(DY+DY0(I))                                          1332
   ISN 0045              DY1(I)=DY                                                              1333
   ISN 0046              GO TO 3000                                                             1334
   ISN 0047      1400    Y0(I)=Y0(I)+H*(DY*A1+DY1(I)*A2+.5D0*DY0(I))                            1335
   ISN 0048              CIND=Y0(I)                                                             1336
   ISN 0049              GO TO 3000                                                             1337
   ISN 0050      1500    CIND=Y0(I)+H*(DY*1.5D0-.5D0*DY0(I))                                    1338
```

```
ISN 0051           DY1(I)=DY                                                     1339
ISN 0052           GO TO 3000                                                    1340
ISN 0053    1600   YO(I)=YO(I)+H*(A3*DY+A4*DY1(I)-A5*DYO(I))                     1341
ISN 0054           CIND=YO(I)                                                    1342
ISN 0055           GO TO 3000                                                    1343
ISN 0056    1700   CIND=YO(I)+H*(A6*DY-A7*DY1(I)+A8*DYO(I))                      1344
ISN 0057           GO TO 2200                                                    1345
ISN 0058    1800   CIND=YO(I)+H*(A9*DY+A10*DY2(I)-A11*DY1(I)+A12*DYO(I))         1346
ISN 0059           GO TO 2000                                                    1347
ISN 0060    1900   CIND=Y+H*(B1*DY-DY2(I)*B2+B3*DY1(I)-B4*DYO(I))                1348
ISN 0061    2000   DYO(I)=DY1(I)                                                 1349
ISN 0062    2100   DY1(I)=DY2(I)                                                 1350
ISN 0063    2200   DY2(I)=DY                                                     1351
ISN 0064    3000   IF (I-50) 4000,5000,5000                                      1352
ISN 0065    4000   I=I+1                                                         1353
ISN 0066           RETURN                                                        1354
ISN 0067    5000   STOP                                                          1355
ISN 0068           END                                                           1356
```

```
       COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=50,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID

ISN 0002            DOUBLE PRECISION FUNCTION CORR (TEL)
              C     PROGRAM TO CALCULATE THE APPARENT RIGHT ASCENSION OF THE SUN
              C     AND THE TIME OF EPHEMERIS TRANSIT.
              C     WRITTEN BY CHAN CROCKER MIT LINCOLN LABORATORY
              C     THIS PROGRAM IS GOOD FOR THE YEAR 1967 AND ACCURATE TO ABOUT
              C     FOUR SECONDS OF ARC ON THE RIGHT ASCENSION AND ABOUT 1.5 SEC
              C     ON THE EPHEMERIS TRANSIT
              C                 DEFINITION OF SYMBOLS
              C     WS = MEAN LONGITUDE OF PERIGEE    PAGE 50 AMER. EPHEMERIS
              C     3.3929 = DAY THAT PERIGEE OCCURS CALCULATED FROM THE TABLE
              C     ON PAGE 50 OF THE EPHEMERIS
              C     105.726712328 = DAY THAT THE APPARENT SOLAR TIME AND
              C     EPHEMERIS TIME ARE THE SAME     = APRIL 15
              C     .985600 = THE DAILY MOTION OF MEAN ANOMALY OF THE SUN
              C     23.44358= INCLINATION OF THE SUN,S ORBIT
              C     ESUN= ECCENTRICITY OF THE SUN,S ORBIT
              C     DAY = DAY OF THE YEAR
              C     RAM = RIGHT ASCENSION OF THE MEAN SUN
              C     23. + 45.6*15./3600. = RIGHT ASCENSION OF THE MEAN SUN ON APRIL 15
              C   V= TRUE AMOMALY    XM = MEAN ANOMALY
              C
              C
ISN 0003            IMPLICIT REAL * 8 (A-H,O-Z)
              C     DUMB IS A DUMMY VARIABLE IN COMMON TO PREVENT AN ERROR MESSAGE
              C
ISN 0004            COMMON/Z8/XLIM  , CONST , XLIMX , CUT   , XNUM  , FUEL  ,
                   1         XX     , WS    , COSI  , ESUN  , SINI  ,DUMB   , JFK
                   2         LLL    , M     , K     , NUM   , I3    , I5
ISN 0005            DAY = TEL / 86400.D0
ISN 0006             IF (I3 .EQ. 7 .OR. I3 .EQ. 8)  GO TO I10
ISN 0008        4   XM = ( .9856D0 * DAY - 3.3929D0) * XX
              C     THE FOLLOWING EXPRESSION FROM MOULTON PAGE  171
ISN 0009            V = XM + 2.D0 * ESUN * DSIN(XM) + 5.D0 * ESUN * ESUN *
                   1                    DSIN(2.D0*XM)  /  4.D0
ISN 0010            U = V + WS
ISN 0011            X = DCOS(U)
ISN 0012            Y = DSIN(U) * COSI
ISN 0013            RA = DATAN2(Y,X)
ISN 0014            IF (RA) 5,6,6
ISN 0015          5 RA = RA + 6.2831853I D0
ISN 0016          6 RAM = (.9856D0*(DAY-105.72671200)+23.D0+45.6D0*15.D0/3600.D0)*XX
ISN 0017            IF(RAM) 7,8,8
ISN 0018          7 RAM = RAM + 6.2831853ID0
ISN 0019          8 RAM =ESMOD(RAM,6.2831853ID0                )
ISN 0020            DIFF = RA - RAM
ISN 0021            IF (DABS(DIFF) - 3.D0) 11,11,12
ISN 0022         12 DIFF = DIFF + 6.2831853ID0
              C      NOTE I3 DOES NOT ATTAIN THE VALUE 4 IN THIS GO TO
              C      I3      ' 2,5                              NO ERROR
```

```
             C       I3       = 3,6                          RANDOM ERROR
             C                 1  2  3  4  5  6
ISN 0023        11  GO TO (20,21,22,20,21,22),I3
ISN 0024        20 CORR = DIFF
                   RETURN
ISN 0025
ISN 0026        22 DIFF = DIFF +  (RAN(D) - .5) * XX
ISN 0027        21 DAY = ESMOD (DAY,365.DD)
             C        WHEN I3 = 5 OR 6   DO A STEP FUNCTION CORRECTION
ISN 0028            IF(I3 .EQ. 5  .OR.  I3 .EQ. 6)           GO TO 30
ISN 0030           IF (DAY-45.D0)100,100,101
ISN 0031       100 VALUE = .08D0 * (DAY + 2.25D0) * XX
ISN 0032            GO TO 31
ISN 0033       101 IF (DAY - 129.D0 ) 102,102,103
ISN 0034       102 VALUE = -.0584D0 * (DAY-111.D0) * XX
ISN 0035            GO TO 31
ISN 0036       103 IF (DAY - 211.D0) 104,104,105
ISN 0037       104 VALUE = .D330D0 * (DAY - 162.D0) * XX
ISN 0038            GO TO 31
ISN 0039       105 IF (DAY - 310.D0) 106,106,107
ISN 0040       106 VALUE = -D.0584D0*(DAY-240.D0)*XX
ISN 0041            GO TO 31
ISN 0042       107 VALUE = .080D * (DAY-363.D0) * XX
ISN 0043            GO TO 31
ISN 0044        30  CALL ALVISE(DAY,VALUE)
ISN 0045        31  CORR    =   DIFF - VALUE
ISN 0046            RETURN
ISN 0047      110   DAY = DMOD (DAY,  365.D0)
ISN 0048            CALL CURT (DAY,VALUE)
ISN 0049            IF (I3 .EQ. 8)  CORR = VALUE * XX + (RAN(D) - .5)*XX
ISN 0051            IF (I3 .EQ. 7)  CORR = VALUE * XX
ISN 0053            RETURN
ISN 0054           END
```

```
      COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=50,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID

  ISN 0002            DOUBLE PRECISION FUNCTION DPNV(Y,DY)                        1438
  ISN 0003            DOUBLE PRECISION Y,DY                                       1439
  ISN 0004            DOUBLE PRECISION CIND                                       1440
  ISN 0005            DPNV=CIND(1,Y,DY)                                           1441
  ISN 0006            RETURN                                                      1442
  ISN 0007            END                                                         1443
```

```
LEVEL 2 FEB 67                          OS/360  FORTRAN H                              DATE  68.067/17.16.59

        COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=50,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID

  ISN 0002          DOUBLE PRECISION FUNCTION FINDV(X,H)                              1444
  ISN 0003          DOUBLE PRECISION X,H                                              1445
  ISN 0004          DOUBLE PRECISION CIND                                             1446
  ISN 0005          FINDV=CIND(0,X,H)                                                 1447
  ISN 0006          RETURN                                                            1448
  ISN 0007          END                                                               1449
```

```
      COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=50,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID
ISN 0002            DOUBLE PRECISION FUNCTION FLAM (XLAM)
              C
              C     THIS FUNCTION COMPUTES A FORCE WHEN THE INPUT PARAMETER IS OUTSID
              C            OF A SPECIFIED BAND.
              C
              C
              C       THE FORCE IS LEFT ON FOR M+1 TIME INTERVALS   ---  THESE ARE
              C       DETERMINED BY THE MAIN PROGRAM AND SET IN COMMON
              C
              C       THE IMPULSES ARE ALLOWED ON ONE DAY INTERVALS, BUT ONCE STARTED
              C            NUM IMPULSES WILL BE GIVEN.
              C
              C       K      K IS A GATE WHICH ALLOWS PASSAGE INTO THE ROUTINE.  IT IS
              C                 SET POSITIVE
              C              USUALLY POSITIVE, BUT CHANGED TO NEGATIVE ONCE PER DAY.
              C              IF IMPULSES ARE BEING GIVEN IT IS LEFT NEGATIVE UNTIL
              C              ALL HAVE BEEN COMPLETED.
              C
              C       XLAM  CURRENT POSITION
              C
              C       XLIM  BOTTOM OF THE BAND
              C       XLIMX TOP OF THE BAND
              C             THUS, XLAM-XLIMX-X .GT. 0 --XLAM IS ABOVE THE BAND
              C                 AND XLAM-XLIM-X .LT. 0 ---XLAM IS BELOW THE BAND
              C
              C       LL     IS CONTROLED BY THE FORCE AND BY THE POSITION RELATIVE TO
              C              THE BAND.  IF LL IS NEGATIVE WE ARE INSIDE THE BAND OR TH
              C              FORCE HAS BEEN COMPLETED.
              C
              C       M      COUNTS THE NUMBER OF IMPULSES BEING ISSUED
              C
              C
ISN 0003           IMPLICIT REAL * 8 (A-H,O-Z)
              C
ISN 0004            LOGICAL NEWFR
              C
ISN 0005            COMMON/FSL/NEWFR
ISN 0006            COMMON/DOVE/XLIMD
              C
ISN 0007           COMMON/Z8/XLIM   , CONST , XLIMX , CUT   , XNUM  , FUEL  ,
                  1          XX     , WS    , COSI  , ESUN  , SINI  , TEL   , JFK   ,
                  2          LLL    , M     , K     , NUM   , I3    , I5
              C
ISN 0008           COMMON/PARAMS/FITHDY,CONSTI, XLIMXD, XDISP, XMASS , CUTBY , A     ,
                  1                F      ,XI     , DL     , XLAMD , TEND  , ALONGR, DATE,
                  2                FORCCT,FORLAM, A1     , A2    , W     , I1    , I2,
                  3                I3T   ,I4    , I5T   , I6    , I7    , NUMIN
ISN 0009           COMMON/SUPSTF/   ABI,    AB2, OFFSET,      I8
              C
```

```
                C
                C
                C      WHEN I3 = 9 SAIL1 IS USED. FORCE IS READ IN FROM SETUPS AS CONST
                C      XLAM IS READ IN FROM SETUPS.
ISN 0010               IF (I3.EQ.9) GO TO 6
                C
ISN 0012              IF  (I3 .EQ. 10) GO TO 3
                C
                C
                C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                C
                C        TEST TO SEE IF THE GATE IS OPEN
ISN 0014            5    IF(K) 8,8,9
                C
                C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                C
                C        THE GATE IS CLOSED
ISN 0015            9 FLAM  =   0.D0
ISN 0016              IF (I8 .NE. 0) GO TO 50
ISN 0018              FORCE=   0.D0
ISN 0019              RETURN
ISN 0020           50 I8    =  3
ISN 0021              CALL CHAN(XLAM)
ISN 0022              FLAM  =  CONST
ISN 0023              RETURN
                C
ISN 0024            6  FLAM = CONST
ISN 0025               RETURN
                C
ISN 0026            3 FLAM  =  CONST
ISN 0027              AB1   =  A1
ISN 0028              AB2   =  A2
ISN 0029              IF (I6.NE.2) GO TO 12
ISN 0031              CALL TACK (XLAM)
ISN 0032              RETURN
ISN 0033           12 CALL CHAN (XLAM)
ISN 0034              RETURN
                C
                C        COMPUTE THE FORCE TO BE APPLIED  -  INCLUDE THE CUT-DOWN.
ISN 0035            8  IF(M .NE. 0) GO TO 4
ISN 0037              CON   =   CONST
ISN 0038              IF(I5 .EQ. 1)   GO TO 21
ISN 0040           42 CONTINUE
ISN 0041              IF(LLL . GT. 0)   CON = CUT*CON
                C
                C        COMPUTE THE CORRECTION FACTOR
                C              NOTE. IF I3 .EQ. 4 THERE IS NO CORRECTION
                C
ISN 0043           21 X     =   0.D0
                C
```

```
              C        MODIFY THE INPUT AS A FUNCTION OF THE CORRECTION                1549
              C              AND DETERMINE WHERE IT LIES RELATIVE TO THE BAND          1550
              C                                                                        1551
ISN 0044            IF(I3 .NE. 4)    X = CORR(TEL)                                     1552
ISN 0046               XTEMP =    XLAM-X                                               1553
              C                                                                        1554
              C  IF WE ARE USING THE OLO FORCE CALCULATIONS JUST CONTINUE.  IF NOT,    1555
              C  GO TO SUBROUTINE NEWFOR                                               1556
              C                                                                        1557
ISN 0047             IF(NEWFR) GO TO 300                                               1558
ISN 0049            IF(XTEMP .GT. XLIMX)    GO TO 11                                   1559
ISN 0051            IF(XTEMP .LT. XLIM)    GO TO 1                                     1560
              C                                                                        1561
ISN 0053             K=1                                                               1562
ISN 0054             GO TO 9                                                           1563
              C                                                                        1564
              C      WE ARE OUTSIDE AND ABOVE, THRUSTING WITH THE ORBIT                1565
ISN 0055        11   FORCE=-CON                                                        1570
ISN 0056            IF (I8.NE.0) I8 = 1                                                1566
ISN 0058            OFFSET = -90.0D0 * XX                                              1567
ISN 0059            AB1    =  A2                                                       1568
ISN 0060            AB2    =  A1                                                       1569
              C       WE ARE OUTSIDE AND BELOW,THRUSTING AGAINST THE ORBIT             1571
ISN 0061             GO TO 4                                                           1576
              C                                                                        1577
              C  IF WE REACH HERE IT MEANS WE ARE USING THE NEW FORCE CALCULATIONS     1578
              C                                                                        1579
ISN 0062       300   CALL NEWFOR(XLIMD,XTEMP,CON)                                      1580
ISN 0063         1   FORCE = CON                                                       1581
ISN 0064            IF(I8 .NE.0) I8 = 2                                                1572
ISN 0066            OFFSET=  0.D0                                                      1573
ISN 0067            AB1    =  A1                                                       1574
ISN 0068            AB2    =  A2                                                       1575
ISN 0069         4  M      =    M + 1                                                  1582
              C                                                                        1583
ISN 0070            IF(I6.EQ.2) CALL TACK(XLAM)                                        1584
ISN 0072        43  IF  (I6 .EQ. 1) CALL CHAN (XLAM)                                   1585
ISN 0074            IF(M .GT. NUM) GO TO 7                                             1586
              C                                                                        1587
              C        WE ARE STILL GIVING THE IMPULSES                                1588
              C                                                                        1589
ISN 0076            FLAM   =    FORCE                                                  1590
ISN 0077            IF (I6.EQ.1) FLAM = DABS (FORCE)                                   1591
ISN 0079            JFK    =    JFK + 1                                                1592
ISN 0080            RETURN                                                             1593
              C                                                                        1594
              C        PUBLISH THE REPORT OF THE FUEL CONSUMED                         1595
              C        WE ARE OUTSIDE AND BELOW                                        1596
              C                                                                        1597
ISN 0081         7   FUEL=FUEL+DABS(CON*XNUM)                                          1598
```

```
ISN 0082          IF (I6 .NE. 0) FUEL = 0.D0
ISN 0084          CORD  =    -X/XX
            C
ISN 0085          WRITE(6,10) JFK,CORD,FORCE,FUEL
ISN 0086       10 FORMAT(1X,I10,'CORD=',1PE15.6,'CON=',1PE15.6,'FUEL USED=',1PE15.6,
                 1'KG')
            C
            C
            C        SET THE GATE TO CLOSED
ISN 0087          K     =    1
ISN 0088          M = 0
ISN 0089          GO TO 9
ISN 0090          END
```

```
LEVEL 2 FEB 67                           OS/360  FORTRAN H                          DATE  68.067/17.17.09

          COMPILER OPTIONS - NAME=  MAIN,OPT=00,LINECNT=50,SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID

   ISN 0002          DOUBLE PRECISION FUNCTION ESMOD (X1,X2)
   ISN 0003          IMPLICIT REAL*8 (A-H,O-Z)
   ISN 0004          IX=IDINT(X1/X2)
   ISN 0005          ESMOD=X1-IX*X2
   ISN 0006          RETURN
   ISN 0007          END
```

REFERENCES

1. M. J. Neufeld and B. M. Anzel, "Synchronous Satellite Stationkeeping," AIAA Paper No. 66-304.

2. J. H. Molitor and M. H. Kaplan, "Optimization of Ion Engine Control Systems for Synchronous Satellites," American Institute of Aeronautics and Astronautics, Summer Meeting, Paper No. 63-273.

3. R. R. Allan and G. E. Cook, "The Long-Period Motion of the Plane of a Distant Circular Orbit," Proc. Royal Society 280, p. 97 (7 July 1964).

4. W. L. Black, M. C. Crocker, and E. H. Swenson, "Stationkeeping a 24-Hour Satellite Using Solar Radiation Pressure," to be published in Journal of Spacecraft and Rockets.

5. L. Blitzer, E. M. Boughton, G. Kang and R. M. Page, "Effect of Ellipticity of the Equator on a 24-Hour Nearly Circular Satellite Orbits," J. of Geophysical Research 67, 329 (1962).

6. C. A. Wagner, "The Drift of a 24-Hour Equatorial Satellite Due To An Earth Gravity Field Through 4th Order," NASA TN D-3317 (1966).

7. C. A. Wagner, "Determination of the Ellipticity of the Earth's Equator from Observations on the Drift of the Syncom II Satellite," NASA TN D-2759 (May 1965).

8. Private communication.

9. Private communication.

10. B. J. Moriarty, "Position Error in Station-Keeping Satellite," Technical Note 1966-21, Lincoln Laboratory, M.I.T. (1 April 1966), DDC 633034.

11. Private communication.

ACKNOWLEDGMENT

Marie Roberts, Eileen Leiber, Chester Small, Harold Feinleib, and William Roberts have written different parts of the SSS program. Walter Morrow, Alvise Braga-Illa, William Black, and Roger Brockett have made some of the suggestions for automatic station-keeping that have been incorporated into the program.

DOCUMENT CONTROL DATA – R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Lincoln Laboratory, M.I.T.	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP None

3. REPORT TITLE

Synchronous Satellite Stationkeeping Simulation

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Technical Note

5. AUTHOR(S) *(Last name, first name, initial)*

Crocker, M. Chandler   Swenson, Ellen H.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
14 May 1968	120	11

8a. CONTRACT OR GRANT NO. AF 19(628)-5167 b. PROJECT NO. 649L c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) Technical Note 1968-10 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)* ESD-TR-68-87

10. AVAILABILITY/LIMITATION NOTICES

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
None	Air Force Systems Command, USAF

13. ABSTRACT

A program has been written to simulate the east-west stationkeeping of a synchronous satellite. Different ways of implementing the thrust sequence of rocket motors and solar sails are discussed.

14. KEY WORDS

synchronous satellites
stationkeeping
solar sailing
orbital mechanics

UNCLASSIFIED
Security Classification